Information Processing and Management 59 (2022) 102746

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

A semantic approach to post-retrieval query performance prediction

Parastoo Jafarzadeh, Faezeh Ensan [*]

Ryerson University, Toronto, Canada

ARTICLE INFO

Keywords:
Query performance prediction
Semantic-enabled prediction
Post-retrieval prediction
Semantic information retrieval

ABSTRACT

The importance of query performance prediction has been widely acknowledged in the literature, especially for query expansion, refinement, and interpolating different retrieval approaches. This paper proposes a novel semantics-based query performance prediction approach based on estimating semantic similarities between queries and documents. We introduce three post-retrieval predictors, namely (1) semantic distinction, (2) semantic query drift, and (3) semantic cohesion based on (1) the semantic similarity of a query to the top-ranked documents compared to the whole collection, (2) the estimation of non-query related aspects of the retrieved documents using semantic measures, and (3) the semantic cohesion of the retrieved documents. We assume that queries and documents are modeled as sets of entities from a knowledge graph, e.g., DBPedia concepts, instead of bags of words. With this assumption, semantic similarities between two texts are measured based on the relatedness between entities, which are learned from the contextual information represented in the knowledge graph. We empirically illustrate these predictors' effectiveness, especially when term-based measures fail to quantify query performance prediction hypotheses correctly. We report our findings on the proposed predictors' performance and their interpolation on three standard collections, namely ClueWeb09-B, ClueWeb12-B, and Robust04. We show that the proposed predictors are effective across different datasets in terms of Pearson and Kendall correlation coefficients between the predicted performance and the average precision measured by relevance judgments.

1. Introduction

The importance of Query Performance Prediction (QPP) for ad-hoc retrieval, which automatically predicts the effectiveness of a given retrieval system for any user query in retrieving and ranking documents, has already been studied in the literature Zhou and Croft (2007). A query performance predictor would help in such cases as re-weighting query terms based on their predicted effects on the retrieval performance (Karisani, Rahgozar, & Oroumchian, 2016; Roitman, 2019b), configuring system settings for different query types (Chifu, Laporte, Mothe, & Ullah, 2018), and selecting *low-risk* queries for query expansion (Amati, Carpineto, & Romano, 2004; Azad & Deepak, 2019). QPP is also essential in the context of semantic search techniques, i.e., retrieval methods that model queries and documents based on their meaning, context, and intent (Dietz, Kotov, & Meij, 2018; Lashkari, Bagheri, & Ghorbani, 2019). Given semantics-based methods have been reported to be more effective for *difficult* queries (Ensan & Bagheri, 2017), QPP methods can help by re-routing a query to a more effective retrieval method when they are estimated to be difficult.

Query performance prediction methods can be classified as either *pre-retrieval*, i.e., they analyze the query and corpus statistics independently of any result list, or *post-retrieval*, i.e., they work based on the analysis of the list of top-ranked documents retrieved in response to a given query by a retrieval system. In *post-retrieval* approaches, which is the focus of this paper, a range of techniques

[*] Corresponding author.
E-mail address: fensan@ryerson.ca (F. Ensan).

https://doi.org/10.1016/j.ipm.2021.102746
Received 12 February 2021; Received in revised form 16 July 2021; Accepted 1 September 2021
Available online 23 September 2021

have been explored in the literature, including those that measure the *clarity* of the search results concerning the collection (Cronen-Townsend, Zhou, & Croft, 2002), analyzing the *robustness* of the result list (Zhou & Croft, 2007), as well as analyzing the retrieval score distribution (Shtok, Kurland, Carmel, Raiber, & Markovits, 2012). Most existing methods (Roitman, 2017; Shtok et al., 2012; Tao & Wu, 2014; Zamani, Croft, & Culpepper, 2018) benefit from information derived from terms and their frequencies in documents, queries, passages, and different versions of ranked documents and collections. However, very few have considered the *semantics* of queries and top-ranked documents for performing query performance prediction.

In this paper, we propose a semantic approach to post-retrieval QPP for ad-hoc retrieval according to which the semantics of a query and its associated top-retrieved documents are analyzed to measure three predictors, namely (1) *semantic distinction*, (2) *semantic query drift*, and (3) *semantic cohesion* for query performance. Our work is based on the crucial *similarity*, *query-drift*, and *cohesion* assumptions adopted by several post-retrieval QPP methods (Raiber & Kurland, 2014; Shtok et al., 2012; Zhou & Croft, 2007). Unlike existing works, we use a semantic approach for defining the relationship between top-ranked documents and queries. Here, we assume that queries and documents are modeled as sets of entities from a knowledge graph, e.g., DBPedia concepts, instead of bags of words. Usually, in existing term-based QPP systems, the similarity between two text pieces is measured based on the exact match of their terms (Shtok et al., 2012; Zhou & Croft, 2007). However, in this context, semantic analysis means measuring semantic similarities of two text pieces based on their representation of knowledge graph entities and their relatedness. The relatedness between entities can be learned from the contextual information represented in the knowledge graph (Feng, Bagheri, Ensan, & Jovanovic, 2017). We measure how semantically similar the retrieved documents are to a given query compared to the whole document collection through semantic distinction. Semantic query drift allows us to estimate how semantically focused the retrieved documents are. Finally, in semantic cohesion, we assess how semantically similar the retrieved documents are to each other.

We evaluate the introduced predictors' performance and a linear combination of them on three collections, ClueWeb09-B, ClueWeb12-B, and Robust04. The former two web collections are known to be challenging datasets for the task of query performance prediction compared to other relatively small TREC collections due to their size and the diversity of their queries and documents (Zamani et al., 2018). We show that our method outperforms the state-of-the-art post-retrieval QPP techniques based on different correlation measures.

The major contributions of this paper are as follows:

- We introduce a set of semantic post-retrieval query performance predictors for ad-hoc retrieval based on measuring semantic similarities of top-ranked documents retrieved by a retrieval model, queries, and the whole collection.
- We empirically evaluate four different models for measuring semantic similarities when employed by proposed predictors, namely the graph-based model, the probabilistic graphical model, the centrality-based model, and the coherence-based model, and thoroughly analyze their impact on performance prediction.
- We conduct a set of experiments and empirically illustrate that the proposed predictors effectively predict the performance of queries across different datasets in terms of Pearson and Kendall correlation coefficients. We also show that these predictors outperform all baselines in predicting the performance of *single-entity queries*, a category of queries focused on only one single entity in a knowledge graph. The category of single-entity queries is shown to be popular in query logs based on the query type analysis research (Laclavík & Ciglan, 2013; Pound, Mika, & Zaragoza, 2010) and is investigated in analyzing retrieval methods before (Fafalios, Kasturia, & Nejdl, 2018).

This paper is organized as follows: The following section reviews related works on pre-retrieval, post-retrieval, and semantic-based query performance prediction methods. Section 3 introduces the semantics approach to post-retrieval QPP and presents semantic post-retrieval query performance predictors for ad-hoc retrieval. Also, in this section, we explain semantic similarity models adopted in defining semantic predictors. In Section 4, we report the experiments we conducted to evaluate the proposed method, provide three research evaluation objectives, and analyze these research objectives based on the reported results. Finally, Section 5 concludes the paper.

2. Related work

This section, first, reviews previous works on query performance prediction, their underlining intuition, and the adopted heuristics. Second, it thoroughly investigates how knowledge embedded in knowledge graphs or extracted from linked open data has been used so far for performance prediction. There is a rich body of research on query performance prediction methods (Bagheri, Arabzadeh, Zarrinkalam, Jovanovic, & Al-Obeidat, 2020; Carmel & Yom-Tov, 2010; Cronen-Townsend et al., 2002; Hauff, 2010; He & Ounis, 2004, 2006; Roy, Ganguly, Mitra, & Jones, 2019; Zamani et al., 2018; Zhao, Scholer, & Tsegay, 2008). Existing methods can be classified into pre-retrieval and post-retrieval approaches. According to the pre-retrieval approach, a predictor measures a query's difficulty before the retrieval stage, mostly based on query terms, context, and corpus statistics (He & Ounis, 2004). On the other hand, a post-retrieval predictor estimates query performance after the retrieval stage by analyzing the retrieved documents' ranked list.

2.1. Pre-retrieval query performance prediction

The pre-retrieval approach has been the focus of numerous query performance prediction works during the last several years (Bagheri et al., 2020; He & Ounis, 2004, 2006; Zhao et al., 2008). One important category of pre-retrieval methods are those based on ***specificity*** intuition, i.e., they assume that the more query terms are specific, the easier it would be to locate the relevant content in the collection and so the easier the query will be. Specificity is usually defined and estimated based on Inverse Document Frequency (IDF) and Inverse Collection Term Frequency (ICTF) measures (He & Ounis, 2004). In He and Ounis (2006), specificity is estimated by measuring the Kullback–Leibler divergence between query and collection language models, where a language model is defined as a multinomial distribution over words.

Another direction of work in the pre-retrieval approach is based on ***similarity*** intuition. Here, a query is assumed to perform better in a collection where the corpus has documents similar to the query, so highly similar questions to the collection would be easier to answer. The SCQ predictor introduced in Zhao et al. (2008) is a sample of these methods that computes the query's similarity to the collection by estimating its TF–IDF score. The query terms' frequency is measured over a large document composed of all of the existing documents' texts in the collection.

Variability is another intuition adopted by some pre-retrieval predictors (Zhao et al., 2008). Here, the assumption is that queries whose term weights in the documents are widely dispersed over the collection are easier to be differentiated by a retrieval system and thus perform better than those whose term weights are closer to the mean.

Finally, ***query term relatedness*** hypothesizes that queries whose terms are *related* to each other are easier to answer. Collection-based and WordNet-based predictors introduced in Hauff (2010) are samples of methods with this intuition, where the relatedness between query terms is estimated based on their co-occurrence in the collection and based on their distance in the WordNet graph respectively.

2.2. Post-retrieval query performance prediction

Post-retrieval methods analyze the ranked list of documents returned by an IR system for estimating how a query performs under that specific retrieval system. Consequently, contrary to the pre-retrieval methods, these methods generate system-dependent predictions for query performance, i.e., while a query may be predicted to be difficult for an IR system, it could be predicted to perform well under another one. Observably, post-retrieval QPP methods can be exploited more effectively than pre-retrieval ones in re-routing queries to appropriate retrieval systems and interpolating different retrieval approaches. In the following, we report on important approaches in post-retrieval performance prediction.

An important direction of work in the post-retrieval category is QPP based on the ***clarity*** of a query. Here, the main idea is that the more focused the top-ranked documents are on the topic of a query, the clearer the query is, and thus, the easier it is to answer it. The clarity predictor introduced in Cronen-Townsend et al. (2002) is a widely-adopted one. It estimates the top-ranked documents' focus on the query topic by measuring the Kullback–Leibler divergence between the query relevance-based language model derived from the top-ranked result document and the collection language model.

The next category of post-retrieval methods focuses on the ***robustness*** of the top-ranked result documents, i.e., how stable the ranked list is in the presence of noise in the retrieval process. Here, the assumption is that the more stable the ranked list is, the more effective the retrieval would be, and so, the query is predicted to be an easy one for the retrieval system. Examples of this class of QPP methods include (1) the predictor introduced in Zhou and Croft (2006), which implements the notion of noise by creating a *noisy* collection of documents and running queries over that, and (2) the *QF* (Zhou & Croft, 2007) predictor that creates new noisy queries and runs them over the original collection. In Zhou and Croft (2006), robustness is quantified by employing Spearman rank correlation coefficient for finding the similarity between the original and the noisy ranked list. In Zhou and Croft (2007), the similarity between the original and noisy ranks is measured by the number of overlapping documents in two lists.

Cohesion and ***query drift*** are the other hypotheses behind a class of post-retrieval QPP methods. *Cohesion* of a list is defined as the degree of relatedness between the documents of the top-ranked list (Raiber & Kurland, 2014). The primary hypothesis in this category is that a cohesive list is more likely to be relevant to a given query than a diverse one. *Query drift* indicates the potential amount of non-relevant texts, aspects, or topics to a given query in the top-ranked result documents. Here, the main assumption is that a low degree of query drift correlates with search effectiveness. The Normalized Query Commitment (NQC) predictor introduced in Shtok et al. (2012) quantifies query drift using the standard deviation of document scores in the top-ranked result list. Some extensions to NQC have been suggested in Roitman (2019a), and their effectiveness across different datasets has been analyzed.

Finally, ***distinction*** hypothesizes that the retrieval effectiveness is positively correlated with the distinction of the top-ranked documents, where *distinction* shows how much more similar the retrieved documents are to the query than the collection. Weighted Information Gain (WIG) (Zhou & Croft, 2007) is among the widely-used predictors in this category that quantifies distinction by measuring the divergence of the mean retrieval score of result documents and the collection score for a given query. Using retrieval scores in the prediction process, NQC and WIG are sometimes categorized in the same class of *score-based* predictors (Shtok, Kurland, & Carmel, 2016).

2.3. Semantics in query performance prediction

A few researchers have explored the use of information from knowledge graphs or linked open data for query performance prediction. In Roy et al. (2019), the query's semantics have been analyzed through embedded word vector representations for queries. These word vectors evaluate query ambiguity by finding the number of clusters of terms that are semantically close to the question. This method (Roy et al., 2019) performs effectively on small-sized collections such as Robust and AP. However, it does not report evaluation results achieved on large-scale web collections such as ClueWeb09-B and ClueWeb12-B. In Bagheri et al. (2020), a set of specificity-based *semantic* pre-retrieval predictors have been introduced. Contrary to existing specificity-based methods (He & Ounis, 2004, 2006), the introduced semantic predictors in Bagheri et al. (2020) quantify specificity in terms of the semantics relatedness between terms measured by the cosine similarity of the vector representation of terms in a neural embedding space.

In Krikon, Carmel, and Kurland (2012) and Raviv, Kurland, and Carmel (2014), entities have been used for performance prediction in tasks other than ad-hoc retrieval. In Krikon et al. (2012), new measures based on entity similarities are proposed that predict retrieval performance in question answering systems. In Raviv et al. (2014), entities are used for defining predictors for *entity ranking*. In Raviv et al. (2014), several pre and post-retrieval baseline predictors have been adopted for entity ranking. The main difference between these works and the predictors presented in this paper is in their objective. These works aim at predicting question answering or entity ranking, i.e., finding and ranking entities that can better represent the intent of queries. Nonetheless, our work's main objective is to predict a retrieval system's performance in retrieving the most relevant documents to a query using semantic information.

The works presented in Chifu et al. (2018) and Zamani et al. (2018) are samples of recent works in QPP literature for ad-hoc retrieval that apply supervised learning methods for prediction. In Chifu et al. (2018), a set of learning to rank features previously defined and applied for document ranking has been employed for performance prediction. In Zamani et al. (2018), a neural network framework for QPP is introduced that has three main components: the first and the second components analyze the retrieval scores and the term distributions of top-ranked documents, respectively. The third component defines document representations based on the embeddings of terms in a l-dimensional space and their *global importance*, and how semantically coherent or diverse are the top-ranked documents. In Déjean, Ionescu, Mothe, and Ullah (2020), a feature selection approach for QPP and its effectiveness for selecting a limited number of features has been introduced.

Since this paper's focus is on semantic-based post-retrieval predictors, we analyze several baseline post-retrieval methods presented in this section (Chifu et al., 2018; Cronen-Townsend et al., 2002; Kurland, Shtok, Carmel, & Hummel, 2011; Shtok et al., 2012; Tao & Wu, 2014; Zamani et al., 2018; Zhou & Croft, 2007) in our experimental evaluation and compare their performance with our method. All experimental results are reported in Section 4.

3. Proposed method

In this section, we first explain the semantics approach for QPP and introduce semantic post-retrieval query performance predictors for ad-hoc retrieval. Afterward, we present the adopted models for measuring semantic similarities between documents and queries.

3.1. A semantic approach for query performance prediction

In the following, we introduce three types of semantics-based post-retrieval predictors based on *distinction*, *query drift*, and *cohesion* hypotheses. The main objective is to estimate the performance of a given query q by a retrieval method as accurately as possible using information extracted from the top-ranked documents, denoted as $\mathcal{D}_q$, when no relevance judgment is available. Here, we assume that queries and top-ranked documents have been linked to sets of a knowledge graph's entities such as DBPedia (Auer et al., 2007).

3.1.1. Semantic distinction

The first predictor is based on the hypothesis that the more semantically similar a given query is to the top-ranked documents compared to the whole collection, the more effective the retrieval would be.

For approximating semantic distinction, we use the notions of *mutual information* and *weighted entropy* that have been exploited in the context of query performance prediction (Zhou & Croft, 2007). Let $\mathcal{Q}_{all}$ be the set of all possible queries that can be posed to a retrieval model, and $\mathcal{D}_{all}$ be the set of all possible documents that can be retrieved by the retrieval model, $P(q)$, for $q \in \mathcal{Q}_{all}$, represents the probability distribution of queries, i.e., how probable it is to get q as an input query. Further, $P(d)$, for $d \in \mathcal{D}_{all}$, represents the probability distribution of documents, the probability of retrieving d independently from the query. In this setting, mutual information is a measure that approximates the amount of information that one random variable (the query's random variable depicted by q) contains about another random variable (the document's random variable depicted by d) (Cover & Thomas, 1991) and is defined as follows:

$$I(q,d) = \sum_{q' \in \mathcal{Q}_{all}} \sum_{d' \in \mathcal{D}_{all}} log\ P(q',d') \tag{1}$$

Simply put, $I(q,d)$ shows how much information about documents can be obtained by observing the queries. This definition of mutual information does not depend on a specific query or the documents retrieved by a retrieval model for that query. The *weighted entropy* concept has been introduced in the literature Kelbert, Stuhl, and Suhov (2017) to make mutual information context-dependent, i.e., it approximates mutual information based on the specific values assigned to q and d (Kelbert et al., 2017). Eq. (2) defines the mutual information between a query q and document d based on the weighted entropy:

$$H(q,d) = \sum_{q' \in \mathcal{Q}_{all}} \sum_{d' \in \mathcal{D}_{all}} \alpha_{q',d'} \, log \, P(q',d') \tag{2}$$

where $\alpha_{q',d'}$ is a non-negative weight that is dependent on the query and documents. Analogous to Zhou and Croft (2007), we choose $\mathcal{D}_{all}$ to be equal to D_q, the set of all documents retrieved by a retrieval model for query q, and α to be zero for all $q' \neq q$ and be $\frac{1}{|D_q|}$ for $q' = q$. This way, we disregard the impact of other queries and the impact of the documents that are not retrieved in response to the query for approximating the mutual information between a query and a set of retrieved documents. In this case, the mutual information between a query q and a document d can be estimated by the following:

$$H(q,d) = -\frac{1}{|D_q|} \sum_{d' \in D_q} log \, P(q,d') \tag{3}$$

Similarly, the mutual information between a query and the whole collection, $H(q,C)$, estimates how much information about a query q can be obtained by observing collection C created by concatenating all documents' content in one document (Zhou & Croft, 2007).

This semantic distinction predictor estimates the difficulty of a query by approximating the difference between two values: first, the mutual information between the query and the whole collection, and second, the mutual information between the query and the retrieved documents. The lower the difference, the less distinct the set of retrieved documents are from the whole collection.

To define semantic distinction, we approximate $P(q,d)$ based on a semantic similarity model that approximates semantic similarities between two pieces of text, e.g., a query q and document d denoted as $SemanticSim(q,d)$. The semantic similarity model is based on the semantic relatedness between entities inside two texts and returns a number between 0 and 1 (Please see Section 3.2 for a comprehensive explanation of adopted semantic similarity models). Here, we formalize $P(q,d)$ to be equal to $e^{SemanticSim(q,d)-1}$. This formalization results in two important characteristics: first, for a case when $SemanticSim(q,d) = 1$ (e.g., when a query is identical to a document), the probability of $P(q,d)$ is equal to 1. Second, the lowest value for $P(q,d)$ is not zero but $1/e$. This means the probability of $P(q,d)$ for a retrieved document d in response to a query q is not zero, even if our semantic similarity models have found no semantic relatedness between the document and the query entities. On this basis, Semantic Distinction predictor, SD, is defined as the difference of the mutual information between the query and the whole collection and the mutual information between the query and the top-ranked documents as follows:

$$SD(q,\mathcal{D}_q,C) = \frac{1}{|\mathcal{D}_q|} \sum_{d' \in D_q} SemanticSim(q,d') - SemanticSim(q,C) \tag{4}$$

The following example illustrates how the introduced semantic approach could be effective when term-based measures fail to quantify this hypothesis correctly.

Let us take the query 'to be or not to be that is the question', TREC Web Track query #70 for the ClueWeb09-B dataset (Callan, Hoy, Yoo, & Zhao, 2009). Consider that the entity representation of this query, when ran through an entity linking system such as Tagme (Ferragina & Scaiella, 2010), is the Wikipedia entry 'To Be or Not to Be', which corresponds to the DBPedia entity with WIKIPAGEID of 3750492 (We refer to it as Entity #3750492). Further, assume we are predicting the performance of the Query Likelihood retrieval model (QL) (Ponte & Croft, 1998). This query is hard for QL (no related documents have been retrieved when ran over the ClueWeb09-B collection). Nonetheless, the top-ranked documents have term-specific features that enable distinction-based measures to predict this query as a simple one. The top-ranked documents have frequent occurrences of the term 'Question', a term that, due to its specificity compared to the other terms in the query, largely contributes to the retrieval score in a term-based retrieval system such as QL. Existing distinction-based QPP predictors such as WIG (Zhou & Croft, 2007) employ the distribution of QL retrieval scores for predicting its performance for a given query, which is ineffective when QL fails to score documents based on their terms correctly.

On the other hand, the semantic distinction-based predictor, which is introduced in this paper, looks at the query entity (Entity #33750492) and the top-ranked documents' entities and considers their semantic relatedness to measure how semantically similar the query is to the top-ranked documents. In this example, the retrieved documents include completely unrelated entities such as International Union of Pure and Applied Chemistry (Entity #14870), Alkene (Entity #2761), and Alcohol (Entity #1014). These unrelated entities enable our proposed predictor to find them indistinct from the collection regarding the query entity and hence correctly predict the query as a hard one.

3.1.2. Semantic query drift

The semantic query drift predictor is based on the estimation of non-query-related aspects of the retrieved documents: the less semantically close a query is to non-relevant aspects of the top-ranked documents, the more effective the retrieval has been for the given query. The primary assumption is that the top-ranked documents might include *misleading* documents, i.e., ones that are not relevant to the query and can negatively impact the retrieval performance (Shtok et al., 2012). To quantify misleading aspects, we find the average of the semantic similarities between documents in the top-ranked list to the query as follows:

$$\hat{\mu}(q) = \frac{\Sigma_{d \in D_q} SemanticSim(q,d)}{|\mathcal{D}_q|} \tag{5}$$

where $SemanticSim(q,d)$ approximates semantic similarities between q and d based on a semantic similarity model and will be explained in Section 3.2.

Here, we assume that $\hat{\mu}(q)$ represents the average semantic relatedness between a query and all topics that the retrieval model has correctly or incorrectly retrieved. On this basis, we introduce the Semantic Query Drift (SQD) predictor that measures how different each document in the top-ranked list is from the average:

$$SQD(q,\mathcal{D}_q,C) = \frac{\sqrt{\frac{1}{|\mathcal{D}_q|}\Sigma_{d\in\mathcal{D}_q}(SemanticSim(q,d)-\hat{\mu})^2}}{SemanticSim(q,C)} \tag{6}$$

where the difference from the average is normalized by $SemanticSim(q,C)$, the similarity of the query to the whole collection, to make the value of the predictor comparable across different queries.

Contrary to SD, SQD is not directly dependent on measuring semantic similarities between each document in the top-ranked list to the query; instead, it measures how different the top-ranked documents are in terms of their semantic similarities to the query. Here, the central assumption is that a retrieval system also returns misleading documents, and the average similarity of documents represents both related and unrelated aspects to a query. According to this assumption, since a low deviation indicates that the documents are close to the average, it is a sign of unsuccessful retrieval.

The following example highlights the importance of our proposed semantics approach in quantifying query drift. Consider the TREC Web Track query #15, ‘espn sports’. This query is a relatively simple one for the QL retrieval model over the ClueWeb09-B collection (It is ranked 33 in a list of 200 TREC Web track queries sorted by their achieved AP over the ClueWeb09-B collection in our experiments). Nonetheless, the retrieved documents contain frequent occurrences of terms ‘espn’ and ‘sport’ that contribute to their relatively high retrieval scores under the QL model, which is based on a term-based language model for retrieval. Consequently, a score-based query-drift predictor such as NQC (Shtok et al., 2012) predicts this query as difficult. The reason is that NQC predictor measures the divergence of the retrieval scores from the average score. Then, it correlates this divergence with query drift and subsequently with search effectiveness. Hence, since the retrieval scores are not highly divergent from the average (any document in the top-ranked list has a relatively good score), the query is predicted difficult.

On the other hand, the introduced semantic-based query drift predictor behaves entirely differently. In this example, the top-ranked documents retrieved by QL contain entities like ‘Association football’, ‘College soccer’, ‘ESPN on ABC’, ‘Ice hockey’, and the ‘The Open Championship’. These entities make them semantically more close to the query than the entities found in the documents in the middle of the list, such as ‘Sponsor (commercial)’, ‘Interview’, ‘Political correctness’, and ‘Spin (propaganda)’. The reason is that our proposed predictors (SQD) quantify the query drift by defining a semantic closeness measure between the top documents and the query and measures the divergence over semantic closeness. Hence, SQD is more effective than term-based predictors when documents are semantically divergent.

3.1.3. Semantic cohesion

The third predictor measures semantic cohesion of top-ranked documents. Here, prediction is based on the hypothesis that the more semantically similar the retrieved documents are, the more coherent the result documents would be. As such, one can assume that a highly cohesive set of documents can be an indicator for more effective retrieval. We use the average semantic similarities between all pairs of documents in the top-ranked documents to indicate semantic cohesion.

$$SC(q,\mathcal{D}_q,C) = \frac{\Sigma_{d\in\mathcal{D}_q,d'\in\mathcal{D}_q}SemanticSim(d,d')}{|\mathcal{D}_q\times\mathcal{D}_q|} \tag{7}$$

where $SemanticSim(d,d')$ is calculated by a semantic similarity model explained in Section 3.2.

It may be observed that the two principles of query drift and cohesion contradict each other. In Shtok et al. (2012), it has been argued that query drift reflects each document’s closeness to the query while cohesion quantifies the similarity of top-ranked documents to each other. Hence, there is no fundamental contradiction. This argument can be disputed considering that a set of highly similar documents to a query can be similar to each other. In this case, two hypotheses predict the performance of a given query differently. This fact highlights the importance of interpolating QPP methods that cover different principles. Section 3.1.4 introduces an interpolation of different predictors, while the interpolation weights are learned over training data. In Section 4.2, we will show that the interpolated approach offers a more stable prediction performance than individual predictors, i.e., it can reach a good performance over all datasets while the performance of its individual components shows high variance.

The following example shows how our proposed semantic approach for quantifying cohesion is more effective than a term-based approach in query performance prediction. Let us take the TREC Web Track query #89, ‘OCD’. This query is a relatively simple one for the QL retrieval model over the ClueWeb09-B collection (ranked 42 in a list of 200 TREC Web track queries sorted by their achieved AP over the ClueWeb09-B collection in our experiments). The top-ranked documents retrieved by QL contain several semantically related terms such as ‘phobia’, ‘mental illness’, ‘behavioural disorder’, ‘drug therapy’, ‘behavioural therapy’, ‘OCD center’, and ‘anxiety center’. A term-based cohesion measure such as Raiber and Kurland (2014) would miss the top-ranked documents’ highly cohesive content because of the mismatch between the retrieved documents’ terms. On the other hand, the introduced semantic-based cohesion predictor would correctly measure the semantic cohesion even in the face of such term mismatch.

3.1.4. Semantic combination

Each of the introduced semantic predictors in Sections Section 3.1.2, 3.1.1, and 3.1.3 is based on a hypothesis covering a specific aspect of query performance prediction. As shown in previous works (Diaz, 2007; Zhou & Croft, 2007), combining different predictors that consider different aspects has a positive effect on predicting the query's performance.

In this paper, for covering different prediction hypotheses, we define SCM, a linear combination of SD, SQD, and SC predictors (Zhou & Croft, 2007), where the coefficient of each predictor in this linear interpolation is learned over the training data. The details of experimental setup for this predictor are reported in Section 4.1.1.

3.2. Semantic similarity models

As illustrated in the previous section, for defining semantic QPP predictors, semantic similarities between a given query and top-ranked documents, semantic similarities between top-ranked documents, and semantic similarities between a query and the collection had to be measured. Estimating semantic similarities have been an active field of research in recent years (Feng et al., 2017; Hussain et al., 2020; Qu, Fang, Bai, & Jiang, 2018). In this section, we explain four methods that we have adopted in this paper. In applying these methods, we assume that documents have been modeled as sets of entities linked to entries in a knowledge graph by an entity linking tool such as Tagme (Ferragina & Scaiella, 2010), Babelfy (Moro, Raganato, & Navigli, 2014), or DBPedia Spotlight (Mendes, Jakob, García-Silva, & Bizer, 2011), and a query is modeled as a short document that comprises a set of entities. Semantic QPP predictors introduced in Section 3.1 are defined independently from the underlying semantic similarity estimator, i.e., they can be measured using any of these four methods or any alternative similarity measure.

3.2.1. Graph-based model

In Graph-Based similarity model, GB, we exploit the topological features of the subset of the knowledge graph represented by documents' entities for measuring documents similarities. The topological characteristics of entities' graphs have been widely used for entity disambiguation, entity matching, and measuring entity distances and relatedness in the literature AlMousa, Benlamri, and Khoury (2021), Ayvaz and Aydar (2019), Hulpuş, Prangnawarat, and Hayes (2015), Li, Xiao, Ma, Jiang, and Zhang (2017), Ma, Alipourlangouri, Wu, Chiang, and Pi (2019) and Traverso, Vidal, Kämpgen, and Sure-Vetter (2016). In this paper, we use hierarchical and non-hierarchical paths between entities to measure entity distances. Inspired by Paul, Rettinger, Mogadala, Knoblock, and Szekely (2016), we define the GB model as comprised of hierarchical and traversal similarity components. The hierarchical components measure the taxonomic distance of two entities based on their Lowest Common Ancestor (LCA) distance in the knowledge graph's taxonomic representation. The traversal model is defined based on the Semantic Connectivity Score (CSS) between any pair of entities e_1 and e_2 introduced in Nunes, Dietze et al. (2013) and Nunes, Kawase et al. (2013) as follows:

$$SCS(e_1, e_2) = \Sigma_{l=1}^{l=\pi} \beta^l |paths^l(e_1, e_2)| \tag{8}$$

where β is a damping factor between 0 and 1, usually set to 0.5, $|paths^l(e_1, e_2)|$ is the number of non-hierarchical paths between e_1 and e_2 in the knowledge graph of length l, π is a parameter for the maximum path length between entities.

More specifically, the graph similarity, $sim_g(e_1, e_2)$, between two entities e_1 and e_2 would be defined according to as the following:

$$sim_g(e_1, e_2) = HSim(e_1, e_2) + TSim(e_1, e_2) \tag{9}$$

where $HSim$ and $TSim$ are hierarchical and traversal similarities defined in Eqs. (10) and (11), respectively.

$$HSim(e_1, e_2) = 1 - \frac{dist(LCA(e_1, e_2), e_1) + dist(LCA(e_1, e_2), e_2)}{dist(root, e_1) + dist(root, e_2)} \tag{10}$$

where $dist(e_1, e_2)$ measures the number of edges between e_1 and e_2 in taxonomy and LCA stands for the lowest common ancestor for two entities.

$$TSim(e_1, e_2) = \frac{\Sigma_{e \in N(e_1) \cap N(e_2)} SCS(e_1, e) * SCS(e_2, e)}{\Sigma_{e \in N(e_1)} SCS(e_1, e) * SCS(e_1, e)} \tag{11}$$

where $N(e_i)$ denotes the entities in the e_i neighborhood, i.e., the entities that are connected to e_i with one edge in the graph.

Based on the introduced measure for entity pairs relatedness, the similarity between two documents d_1 and d_2, which are represented as sets of entities, can be calculated as follows:

$$Sim_{GB}(d_1, d_2) = \frac{\Sigma_{e_1 \in d_1} (\underset{e_2 \in d_2}{Max}\, sim_g(e_1, e_2))}{|d_1| + |d_2|} \tag{12}$$

where $|d_1|$ and $|d_2|$ represent the number of entities in d_1 and d_2, respectively.

3.2.2. Probabilistic graphical model

In the Probabilistic Graphical Model, PGB, we assume a graphical representation of document entities, where weighted edges connect related entities in the knowledge graph in the graph. Inspired by Ensan and Bagheri (2017), we calculate the conditional probability of observing entities in the query given entities in the document to measure semantic similarity. The graph-based similarity model, Sim_{GB} in Section 3.2.1, mainly focuses on estimating similarities between entity pairs based on the knowledge graph topological characteristics. However, PGB gets the relatedness between entity pairs as input, which could be calculated by any similarity estimation method (Gabrilovich, Markovitch, et al., 2007; Strube & Ponzetto, 2006; Yamada et al., 2020), and estimates document similarities on that basis.

More succinctly, a graphical representation of documents and queries can be represented as an undirected graph $G(V,E)$, where $V = V_D \cup V_Q$, and V_D is a set of variables correspond to entities in documents and V_Q is a set of variables corresponds to entities in queries. These variables take integer values 0 or 1, where one indicates that the entity exists in a given document or a given query. Assuming that $N = e_1, e_2, \dots, e_n$ be the set of all the knowledge graph entities, V_D and V_Q can have up to $|N|$ variables. The variables are connected by weighted edges representing semantic relatedness between entity pairs. Hence $(e_q, e_d) \in E$ denotes some degree of semantic relatedness between a query entity e_q and a document entity e_d. Based on this representation, for any document d that is a set of m entities $\langle e_1, \dots, e_m \rangle$, m variables in V_D, corresponding to $e_1, ..e_m$, will get the values of 1 and the others get the value of 0. Here, the core idea is to measure similarities between a query and a document based on the conditional probability of a set of query variables, given the set of observed variables in a document.

In other words, given that q is a query and $\{e_i \in q\}$ is the set of query variables whose value are 1 based on the entities in q, the similarity between q and d is defined as follows:

$$\begin{aligned} Sim(q,d) &= P(q|d) \\ &\simeq \Sigma_{e_i \in q} log P(e_i|d) \end{aligned} \tag{13}$$

where the entities are assumed to be independent.

We adopted the conditional probability model defined in Ensan and Bagheri (2017) to define $P(e_i|d)$ as follows:

$$P(e_i|d) = \frac{1}{Z(d)} exp(\Sigma_k f_k(Clq_k, e_i, d)) \tag{14}$$

where f_k is a feature defined over the clique Clq_k in the graphical representation of documents and queries and $Clq_k \nsubseteq V_D$. $Z(d)$ is a normalization constant that is assumed to have a uniform distribution for reducing the computational cost.

For simplification, we assume only two-node cliques in the graph, i.e., features are defined over two nodes $e_i \in q$ and $e'_j \in d$, where $(e_i, e'_j) \in E$ (there is an edge between e_i and e'_j in the graphical representation), i.e. the values of the features over all other possible cliques is zero. Here, $f_j(Clq_j, e, d)$ will be defined as follows:

$$f_k(Clq_k, e_i, d) = \Sigma_{e'_j \in d} \frac{fq(e_i) fq(e'_j) SemanticRel(e_i, e'_j)}{\Sigma_{c \in C} \Sigma_{c' \in C} fq(c) fq(c') SemanticRel(c, c')} \tag{15}$$

where Clq_d is a clique on two nodes corresponding to $e_i \in q$ and $e'_j \in d$, $fq(e_i)$ denotes the frequency of e_i entity in the query, $fq(e'_j)$ denotes the frequency of e'_j in the document, $C \subset N$ is the collection of all entities observed in the whole collection of documents and queries, c, and c' are two entities in C, and $fq(c)$, $fq(c')$ denote the frequency of c, and c' in C, respectively. Furthermore, *semanticRel* is an estimation of the semantic relatedness between two knowledge graph entries obtained by a third-party semantic similarity estimation system. We will talk about the adopted semantic similarity estimation between entity pairs in Section 4.1.

In Eq. (15), we normalized the similarity between a pair of entities from a document and a query to the sum of similarities exist between the collection entities.

Finally, based on Eqs. (13)–(15), the similarity between a query and a document can be estimated as follows:

$$Sim_{PGB}(q,d) = \Sigma_{e_i \in q} \Sigma_{(e'_j, d) \in E} \frac{fq(e_i) fq(e'_j) SemanticRel(e_i, e'_j)}{\Sigma_{c \in C} \Sigma_{c' \in C} fq(c) fq(c') SemanticRel(c, c')} \tag{16}$$

In this section we defined the similarity between a query and a document. Assuming that a query is itself a document, this definition can be employed for defining similarities between two documents.

3.2.3. Centrality-based model

In the Centrality-Based Model, CB, we estimate the document similarities based on two factors: first, how similar the document entities are, and second, how *central* each entity is in the document. This model's intuition is that similarities between unimportant or non-central entities in two documents should not contribute to the similarity score. On the other hand, any relatedness between central entities of two documents should be considered with a higher importance (Ni et al., 2016).

Assuming that the relatedness between any pair of entities is given (Please look at Section 4.1), we define CB similarity between two documents to be a weighted average of document entity similarities. The weight of the similarity between two entities from two documents is determined based on each entity's centrality in its associated document.

Table 1
Test collections and Topics.

Collection	Topics	#Documents
ClueWeb09-B	1–200	50,220,423
ClueWeb12-B	201–250	52,343,021
Robust04	301–450 & 601–700	528,000

Table 2
Entities Statistics per query and document.

	#queries	#queries without entities	Average #links per query	#retrieved documents	#retrieved documents without entities	Average #links per document
ClueWeb09-B	200	5	1.26	2000	122	133.7
ClueWeb12-B	50	2	1.54	500	9	138.6
Robust-04	250	25	1.93	2500	133	152.9

Inspired by Ni et al. (2016), we define CB similarity between two documents d_1 and d_2, as follows:

$$Sim_{CB}(d_1,d_2) = \frac{\sum_{e_1 \in d_1} (cent(e_1,d_1) \cdot \underset{e_2 \in d_2}{Max}\ SemanticRel(e_1,e_2))}{\sum_{e_1 \in d_1} cent(e_1,d_1)} + \frac{\sum_{e_2 \in d_2} (cent(e_2,d_2) \cdot \underset{e_1 \in d_1}{Max}\ SemanticRel(e_1,e_2))}{\sum_{e_2 \in d_2} cent(e_2,d_2)} \tag{17}$$

where $cent(e_i,d_i)$ denotes how central entity e_i in document d_i is, and $SemanticRel(e_1,e_2)$ indicates the similarity between a pair of entities provided by a third-party method (Gabrilovich et al., 2007).

There are some methods in the literature that define and quantify the centrality of an entity within a document (Ni et al., 2016). In this work, for a given entity e and a document represented as a set of entities $d = \{e_1, e_2, .., e_n\}$, we define centrality as a measure for estimating the semantic similarity of document entities to the given entity. In other words, $cent(e,d)$ is defined as follows:

$$cent(e,d) = \frac{1}{|d|} \sum_{e_i \in d} SemanticRel(e,e_i) \tag{18}$$

where $|d|$ denotes the number of entities in d.

3.2.4. Coherence based model

The Coherence-Based Model, ChB, is defined based on the Coherence between an entity and its surrounding text in the document, i.e., the similarity of coherent entities with their surrounding text contributes more to the final similarity score than other entities.

We define ChB in Eq. (19) with this critical characteristic that our formulation, similar to PGB and CB methods, allows for employing any third-party method that estimates entity pair relatedness ($SemanticRel(e_1,e_2)$) for any two entities e_1 and e_2.

$$Sim_{ChB}(d_1,d_2) = \frac{1}{|d_1|} \sum_{e_1 \in d_1} \sum_{e_2 \in d_2} \rho_{e_1,d_1} \cdot SemanticRel(e_1,e_2) + \frac{1}{|d_2|} \sum_{e_2 \in d_2} \sum_{e_1 \in d_1} \rho_{e_2,d_2} \cdot SemanticRel(e_1,e_2) \tag{19}$$

where ρ_{e_i,d_i} denotes the coherency of entity e_i with its surrounding text in d_i. Similarly to Ni et al. (2016), we use the semantic coherency measure provided by Tagme (Ferragina & Scaiella, 2010) entity linking system for estimating ρ_{e_i,d_i}.

Given an entity e in document d_1, ChB finds all related entities from d_2 and uses their similarities for calculating document relatedness. PGB, introduced in Section 3.2.2, also has the same approach while CB and GB find the most similar entity in d_2 to e and use their relatedness for estimating document semantic similarity.

4. Experiments

In this section, we report the experiments we performed to evaluate our proposed QPP's effectiveness for predicting a retrieval method's performance in retrieving related documents to queries using information extracted from the top-ranked documents. Through our experiments, we attempt at analyzing the following three research objectives:

- **Research Objective 1: Performance across different datasets** We analyze whether the introduced semantic predictors effectively predict the performance of queries across various datasets, comparing to existing state-of-the-art methods. Besides, we examine whether the proposed semantic predictors have a complementary impact on existing term-based methods, i.e., they exhibit better performance when interpolated with baseline methods.

- **Research Objective 2: The impact of semantic similarity models on semantic predictors** We investigate semantic similarity models' effects on the introduced semantic QPP methods' performance. Here, the main question is which semantic similarity model leads to more effective performance in predicting easy, moderate, and hard queries in our experiment datasets.
- **Research Objective 3: Prediction performance for single-entity queries** Given that a query can be linked to one or more entities, we study which of the proposed semantic predictors would be most effective for determining the difficulty of different types of queries, i.e., single-entity or multiple entities.

In the following, we first report our experimental setup, including the predictor names and their interpolations, datasets, and baselines. We then present our analysis on the analytical objectives by reporting experimental results.

4.1. Experimental setup

In our experiments, we adopted three collections: (1) ClueWeb09-B (Callan et al., 2009) (the TREC Category B, which is the first 50 million English pages of the ClueWeb09 corpora), (2) ClueWeb12-B (the TREC 2013 Category B subset of the ClueWeb12 corpora) (he clueweb12 dataset: Dataset details, 2021), and (3) Robust04 (Voorhees et al., 2005). Table 1 shows the statistics for the datasets used in this evaluation. Following prior work on post-retrieval QPP, we predict the performance of the Query Likelihood model (QL) which has been used in most post-retrieval predictors (Chifu et al., 2018; Cronen-Townsend et al., 2002; Kurland et al., 2011; Shtok et al., 2012; Tao & Wu, 2014; Zamani et al., 2018; Zhou & Croft, 2007) with Dirichlet prior smoothing ($\mu = 1500$). We used the Galago version 3.6 (Croft, Metzler, & Strohman, 2010) implementation for QL. Before indexing the collections, we first filtered out spam documents using the Waterloo spam scorer (Cormack, Smucker, & Clarke, 2011) with a threshold of 60%.[1] We removed stop words from documents using the default list of stop words in Lucene version 7.7.1 (Białecki, Muir, Ingersoll, & Imagination, 2012). We employed Porter Stemmer for stemming document words.

In this paper, we assume that documents and queries are represented as sets of entities. We use Tagme (Ferragina & Scaiella, 2010) for linking queries and documents to Wikipedia entries. It has been shown that Tagme has a reasonable performance on different datasets and over various entity linking and entity disambiguation benchmarks (Cornolti, Ferragina, & Ciaramita, 2013). That is why Tagme has been applied for extracting entities in the pre-processing phase of a number of information processing and information retrieval tasks for finding entities, and also disambiguation entities, i.e., selecting the most relevant candidate from the multiple senses based on the context information. In Zhao, Xiong, Qian, and Boyd-Graber (2020), Tagme is used to tag human authored questions with entities in the task of *factoid question answering*. In Zou, Chen, and Kanoulas (2020), Tagme is employed for linking an item description to entities to form the most important characteristics of an item in the task of recommender systems (Zou et al., 2020). In Al-Khatib et al. (2020), Tagme is used for finding concepts of *claims* in forming an argumentation knowledge graph. And as the last example, Liu, Hua, and Zhou (2021) applied Tagme for annotating text for temporal knowledge extraction.

In this paper, we use the infobox of English abstracts of DBPedia version 2014 (released on 08-Sep-2014) (Auer et al., 2007) as our reference knowledge graph.[2] We use Tagme's API for entity linking and use its recommended confidence value of 0.1 for filtering unreliable entity links (Ferragina & Scaiella, 2010).[3] Table 2 shows the average number of extracted entities per query and per document by Tagme. As shown in Table 2, 2.5% of queries executed on ClueWeb09-B, 4% of queries executed on ClueWeb12-B, and 10% of queries executed on Robust04 have been linked to no entities by Tagme. We exclude them from our experiments. We also exclude the documents in the top-ranked list of documents that have no entities, which include 6.1% of the ClueWeb09-B, 1.8% of the ClueWeb12-B, and 5.3% of the Robust04 dataset.

Although the general-purpose linked open data and their corresponding knowledge graphs, such as DBPedia, usually provides a good coverage, the integration of entity-based and keyword-based models would collectively yield to a better prediction performance, because of a wider range of indicators used for the prediction (both terms and entities), especially when parts of queries have not been linked to entities or linked to wrong entities due to entity linking tool's errors. We evaluated a linear combination of the proposed semantic predictors and some of the state-of-the-art text-based predictors and reported the performance results.

For calculating semantic relatedness between entities ($SemanticRel$ in Eqs. (16), (17), and (19)), we use Explicit Semantic Analysis, ESA (Gabrilovich et al., 2007), one of the reliable semantic relatedness methods (Gabrilovich et al., 2007), trained on Wikipedia. We make the entity representation for the queries and documents publicly available.[4]

To identify statistically significant improvements over baselines, we use paired t-test with Bonferroni correction between two lists of predicted performance for queries produced by a proposed predictor and a baseline and report the predictor as statistically significant than the baseline if $p\text{-}value < 0.05$.

[1] http://plg.uwaterloo.ca/∼gvcormac/clueweb09spam/.

[2] https://downloads.dbpedia.org/2014/en/.

[3] https://sobigdata.d4science.org/web/tagme/tagme-help.

[4] https://github.com/ParastooSJ/SemanticQPP.

4.1.1. Proposed predictors

In Sections Section 3.1.1, 3.1.2, and 3.1.3, we introduced SD, SQD, and SC predictors. All these predictors are dependent on $SemanticSim$, which can be calculated based on one of the semantic similarity models introduced in Section 3.2, namely Sim_{GB}, Sim_{PGB}, Sim_{CB}, or Sim_{ChB}. This section evaluates the performance of four variations made for SD, namely SD_{GB}, SD_{PGB}, SD_{CB}, and SD_{ChB}. It also evaluates four variations for SQD, namely SQD_{GB}, SQD_{PGB}, SQD_{CB}, and SQD_{ChB}, and four variations of SC with names SC_{GB}, SC_{PGB}, SC_{CB}, and SC_{ChB} that corresponds to the choice of $SemanticSim$.

We also evaluate combined predictors SCM introduced in Section 3.1.4. Similar to previous work (Zamani et al., 2018; Zhou & Croft, 2007), we randomly split queries of an experimental setting into two equal-sized sets, train, and test sets to learn the interpolation weights. A two-fold cross-validation strategy is used to learn the combined predictors' weights by minimizing the error between the train set's predicted values and their average precision AP(@1000). The described procedure is repeated 30 times, and we report the average score over all of the splits as the final evaluation score. Based on the semantic similarity model choice, we have four SCM variations, i.e., SCM_{GB}, SCM_{PGB}, SCM_{CB}, and SCM_{ChB}. Since our predictor provides an entity-based model for query performance prediction, combining them with a term-based approach could significantly improve effectiveness. In this paper, we report the performance of $SCM + WIG$, $SCM + Clarity$, and $SCM + UEF_{Clarity}$, which interpolates SD, SQD, and SC predictors with WIG (Zhou & Croft, 2007), Clarity (Cronen-Townsend et al., 2002), and UEF (Kurland et al., 2011), respectively.

4.1.2. Baselines

Our baselines include $Clarity$ (Cronen-Townsend et al., 2002), which is based on the KL divergence between the highly-ranked documents' language model and the collection language model; the Normalized Query Commitment (NQC) (Shtok et al., 2012), Weighted Information Gain (WIG) (Zhou & Croft, 2007), and Score Magnitude and Variance (SMV) (Tao & Wu, 2014), which are based on the analysis of the retrieval scores, Query Feedback (QF) (Zhou & Croft, 2007), which is based on the robustness of the retrieval results; and the utility estimation framework (UEF) introduced in Kurland et al. (2011). We also compare our work against two supervised methods reported in Zamani et al. (2018), NeuralQPP (Pairwise) and WMODEL (Chifu et al., 2018) and a pre-retrieval neural embedding method introduced in Bagheri et al. (2020). To make the evaluation results comparable, we use the implementations and the parameter settings provided in Zamani et al. (2018) for the baselines. For $WMODEL$, we report the most effective predictor, $WMODEL : DFIZ_s$, where the number of top-ranked documents is set to 1,000. For neural embedding method (Bagheri et al., 2020) we report the result of $MaxPR$ and $AvgIEF$ predictors. For $NeuralQPP$, $MaxPR$, $AvgIEF$, and $WMODEL : DFIZ_s$ no evaluation results have been reported for ClueWeb12-B and ClueWeb09-B. In our work, we set the number of top-ranked documents to 10.

4.2. Results

This section reports the experimental results on the proposed predictors' performance and compares them with the baselines. In line with other related work in the literature Zhou and Croft (2007), we report Pearson ρ and Kendall τ coefficients for the correlation between the list of queries' predicted performance by the introduced predictors and the list of average precision for the top 1000 documents retrieved per query (AP@1000) based on the collection relevance judgments in Table 3. Pearson correlation reports the correlation between the list of the predicted values by a predictor for all of the queries in a dataset and the list of average precision (AP@1000) for those queries, while Kendall reports the correlation between these two lists based on the rank of queries in each list, when two lists are formed as follows: one list comprises of queries that are ranked from too easy to too difficult based on a given predictor and the other list comprises of the same queries ranked based on their AP@1000. The higher correlation value indicates the better performance of the proposed predictor. For example, the Kendall's tau of 1 means the predictor ranked queries from easy to difficult exactly the same as if these queries were ranked based on their AP@1000.

It can be seen in the table that SC_{GB}, SD_{CB}, SC_{CB}, SCM_{GB}, SCM_{CB}, $SCM_{PGB}+Clarity$, $SCM_{GB}+Clarity$, $SCM_{CB}+Clarity$, $SCM_{GB} + WIG$, $SCM_{CB} + WIG$ and also $SCM + UEF_{Clarity}$ predictors for PGB, GB, and CB variations outperform all predictors on the ClueWeb09-B dataset in terms of both Kendall and Pearson correlations. Also, SD_{PGB}, SD_{CB}, SCM_{PGB}, SCM_{ChB}, $SCM_{PGB} + Clarity$, $SCM_{ChB} + Clarity$, $SCM_{PGB} + UEF_{Clarity}$, $SCM_{PGB} + WIG$, and $SCM_{ChB} + WIG$ outperform all baselines on the ClueWeb12-B datasets on both Kendall and Pearson correlations. On Robust04 dataset, SC_{GB}, SC_{CB}, SCM_{PGB}, SCM_{GB}, $SCM_{PGB}+Clarity$, $SCM_{GB}+Clarity$, $SCM_{PGB}+UEF_{Clarity}$, $SCM_{GB}+UEF_{Clarity}$, $SCM_{PGB}+WIG$, and $SCM_{GB}+WIG$ show a better performance than all baselines except NeuralQPP (Zamani et al., 2018), which is a neural network model learned over three components, namely ranked documents' scores, documents coherency, and documents' distribution.

A notable remark about the reported results is that most of the previous post-retrieval query performance predictors have not reported evaluation experiments for both ClueWeb09-B and ClueWeb12-B datasets (e.g., Chifu et al., 2018; Raiber & Kurland, 2014; Zamani et al., 2018). The ones that reported their results on these datasets exhibit a poor or moderate performance over one dataset while achieving a good performance over the other (e.g., WIG and SMV). On the other hand, our results show that the proposed semantic predictors $SCM_{PGB}+Clarity$ and $SCM_{PGB}+UEF_{Clarity}$, reach a prediction performance on two large-scale ClueWeb09-B and ClueWeb12-B Web document collections that have not been gained by any state-of-the-art systems before.

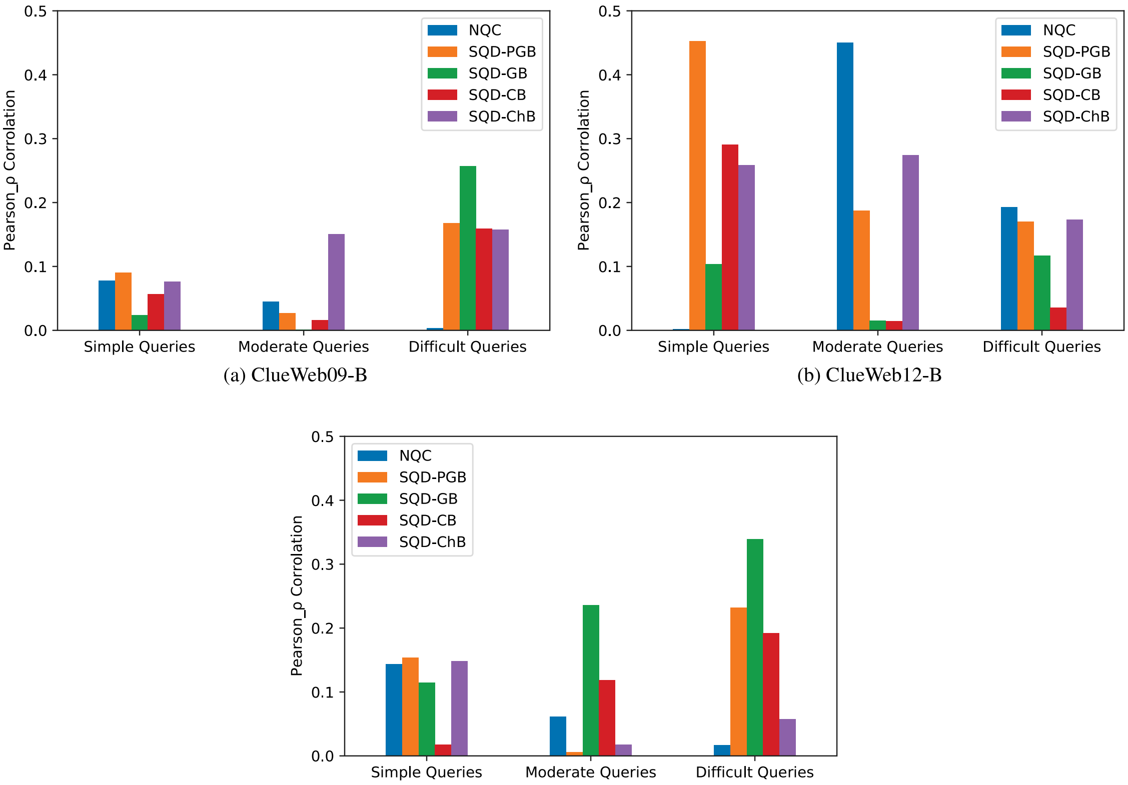
Table 3 confirms the realization reported by earlier research Chifu et al. (2018) and Zamani et al. (2018) that the individual predictors' performance, with no combination, exhibits high variance. For example, while SC_{CB} and SC_{GB} achieve the highest prediction performance over Robust04 and ClueWeb09-B datasets, they perform poorly on the ClueWeb12-B dataset. On the other hand, SD_{CB}, whose prediction performance outperforms all baselines over the ClueWeb12-B dataset, exhibits a moderate performance over the Robust04 dataset.

Table 3
Pearson and Kendall correlation results of query performance predictors on TREC TOPICS 1–200 on ClueWeb09-B, 201–250 on ClueWeb12-B, 301–450 and 601-700 on Robust04. All reported values of the proposed predictors are statically significant compared to all baselines ($p\_value < 0.05$). Bold numbers show the proposed metrics values that are higher than all baselines in each column.

	Method	ClueWeb09-B		ClueWeb12-B		Robust04	
		Pearson ρ	Kendall τ	Pearson ρ	Kendall τ	Pearson ρ	Kendall τ
Baselines	Clarity	0.257	0.221	0.316	0.192	0.170	0.190
	NQC	0.078	0.057	−0.134	−0.044	0.409	0.333
	WIG	0.235	0.167	0.040	0.011	0.008	0.077
	SMV	0.148	0.144	0.220	0.235	0.334	0.300
	QF	0.152	0.075	−0.017	−0.046	0.386	0.307
	$\text{UEF}_{Clarity}$	0.294	0.253	0.271	0.187	−0.065	0.009
	UEF_{WIG}	0.272	0.179	0.155	0.093	0.274	0.250
	WMODEL : DFIZ_s	–	–	0.302	–	0.402	–
	MaxPR	0.340	0.210	–	–	0.360	0.260
	AvgIEF	0.270	0.230	–	–	0.420	0.370
	NeuralQPP	0.367	0.229	–	–	0.611	0.408
Probabilistic Graphical -based	SQD_{PGB}	0.283	0.137	0.275	0.185	0.333	0.247
	SD_{PGB}	0.274	0.234	0.316	0.211	0.404	0.337
	SC_{PGB}	−0.109	−0.084	−0.128	−0.097	0.466	0.298
Graph based	SQD_{GB}	−0.105	0.07	0.048	0.071	0.201	0.110
	SD_{GB}	0.280	0.231	0.170	0.234	0.482	0.322
	SC_{GB}	**0.412**	0.220	−0.034	0.039	0.525	0.384
Centrality -based	SQD_{CB}	−0.240	−0.102	−0.112	−0.068	−0.303	−0.182
	SD_{CB}	**0.379**	**0.288**	**0.397**	**0.261**	0.333	0.262
	SC_{CB}	**0.397**	0.237	−0.002	0.114	0.514	0.362
Coherence -based	SQD_{ChB}	−0.094	−0.050	−0.217	−0.061	0.137	0.091
	SD_{ChB}	0.282	**0.279**	0.257	0.227	0.266	0.260
	SC_{ChB}	−0.059	−0.040	−0.233	−0.249	0.144	0.189
Semantic Combination	SCM_{PGB}	0.312	0.221	**0.462**	**0.329**	0.531	0.367
	SCM_{GB}	**0.385**	0.232	−0.122	−0.041	0.567	0.381
	SCM_{CB}	**0.422**	**0.270**	0.220	0.206	0.507	0.302
	SCM_{ChB}	0.254	**0.256**	**0.342**	**0.249**	0.156	0.220
Semantic Combination +Clarity	$\text{SCM}_{PGB} + Clarity$	**0.372**	**0.257**	**0.478**	**0.323**	0.557	0.362
	$\text{SCM}_{GB} + Clarity$	**0.388**	**0.260**	0.118	0.124	0.587	0.370
	$\text{SCM}_{CB} + Clarity$	**0.427**	**0.295**	0.271	0.160	0.488	0.273
	$\text{SCM}_{ChB} + Clarity$	0.300	0.246	**0.323**	**0.240**	0.166	0.181
Semantic Combination +$\text{UEF}_{Clarity}$	$\text{SCM}_{PGB} + UEF_{Clarity}$	**0.386**	**0.268**	**0.441**	**0.331**	0.531	0.356
	$\text{SCM}_{GB} + UEF_{Clarity}$	**0.411**	**0.270**	0.096	0.157	**0.620**	0.391
	$\text{SCM}_{CB} + UEF_{Clarity}$	**0.444**	**0.300**	0.284	0.218	0.502	0.317
	$\text{SCM}_{ChB} + UEF_{Clarity}$	0.319	**0.256**	0.282	0.220	0.251	0.298
Semantic Combination +WIG	$\text{SCM}_{PGB} + WIG$	0.342	0.251	**0.494**	**0.341**	0.536	0.361
	$\text{SCM}_{GB} + WIG$	**0.412**	0.237	−0.080	−0.057	0.590	0.378
	$\text{SCM}_{CB} + WIG$	**0.409**	**0.276**	0.240	0.192	0.480	0.272
	$\text{SCM}_{ChB} + WIG$	0.270	0.236	**0.350**	**0.291**	0.116	0.144

It can be observed from Table 3 that predictors SCM_{PGB}, SCM_{CB}, and SCM_{ChB} which are formed based on a linear interpolation of other semantic predictors, show a more stable prediction performance. For example, according to Table 3, SCM_{CB} has a Pearson ρ of 0.488 on Robust04 dataset, which is less than the reported Pearson ρ of 0.514 for one of its components SC_{CB} over the same dataset. On the other hand, SCM_{CB} achieves a Pearson ρ of 0.271 on ClueWeb12-B dataset, making it a good predictor among baselines and other semantic predictors, especially compared to SC_{CB} achieves a Pearson ρ of −0.002 on the same dataset. Observably, the simple linear interpolation of semantic predictors contributes to a more stable prediction across different datasets. We leave the study of other interpolation approaches other than linear interpolation to gain a better prediction performance for future works. Table 3 also shows that the interpolation of $Clarity$, WIG, and $UEF_{Clarity}$ with semantic-based predictors is boosting their prediction performance mainly on the ClueWeb09-B dataset (e.g., $SCM_{PGB} + Clarity$ has a better Pearson ρ and Kendall τ coefficients than SCM_{PGB} on ClueWeb09-B dataset).

Based on the results depicted in Table 3, a linear combination of predictors based on the PGB semantic similarity model (SCM_{PGB}, $SCM_{PGB} + Clarity$, $SCM_{PGB} + UEF_{Clarity}$, and $SCM_{PGB} + WIG$) show more consistent performance over different datasets, compared to the predictors defined based on other similarity models. The performances of $SCM_{PGB} + Clarity$ and $SCM_{PGB} + UEF_{Clarity}$ are solid on both ClueWeb09-B and ClueWeb12-B datasets and outperform all baselines. Their performance on Robust04 is among the top-three ones, i.e., outperforming all baselines except $NeuralQPP$ regarding both Pearson and Kendall coefficients. On the other hand, the $SCM + WIG$ with GB similarity model shows a poor performance on ClueWeb12-B. Also, $SCM + WIG$ with the ChB similarity model exhibits a moderate performance across all datasets, especially Robust04.



(a) ClueWeb09-B

(b) ClueWeb12-B

(c) Robust4

Fig. 1. Pearson correlation between simple, moderate, and hard queries' AP and estimated values by four semantic query drift (SQD) predictors, and Normalized Query commitment (NQC) predictor on ClueWeb09-B, ClueWeb12-B, and Robust4. In the plot, the X-axis represents the queries' category and Y-axis represents Pearson correlation values.

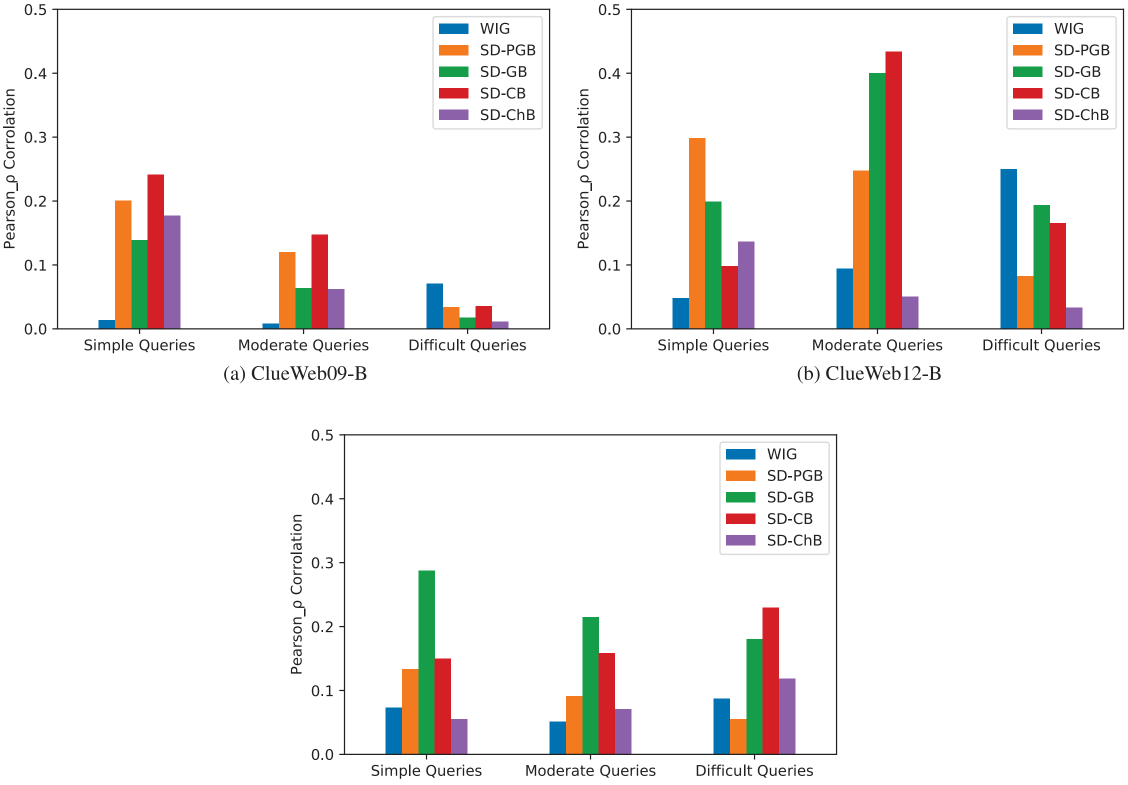
4.3. Predictors performance based on semantic similarity models

Section 3.2 introduces four different semantic similarity models: graph-based model, probabilistic graphical model, centrality-based model, and coherence-based model, abbreviated by GB, PGB, CB, and ChB, respectively. This section analyzes different predictors' performance when exploiting these semantic similarity models. In the following, we first analyze the semantic similarity model's impact on SQD, SD, and SC predictors.

4.3.1. Semantic query drift (SQD) predictors

To analyze the performance of semantic query drift (SQD) predictors, we classify the queries of each dataset into three categories: simple queries, queries with an Average Precision (AP) of 0.2 or more; difficult queries, queries with an AP less than 0.1; and moderate queries, queries with an AP between these two values. We report the Pearson correlation coefficient ρ between the predicted performance by semantic query drift (SQD) predictors in each category and the average precision for the top 1000 documents retrieved per query (AP@1000) based on the collection relevance judgements. Fig. 1 shows the Pearson correlation coefficient ρ for each category over ClueWeb09-B, ClueWeb12-B, and Robust04 datasets. We also report the Pearson correlation coefficient for NQC in addition to SQD predictors, a term-based predictor based on the query drift hypotheses (Please see section 2.2 for more details on NQC). As we can see in this figure, SQD_{PGB} achieves a higher correlation than other predictors on simple queries over all three datasets. In the other words, SQD_{PGB} can most-correctly classify simple-queries as those with the highest predicted APs on all three datasets. Furthermore, SQD_{PGB} shows a fair performance on difficult queries. The main weakness of this algorithm falls on moderate queries where the ρ value is among the three-lowest values compared with other predictors. Semantic query drift is based on this hypothesis that the more semantically distant each retrieved document is from the average similarity of the retrieved documents to the query, the easier the query would be. Fig. 1 shows that SQD_{PGB} is on average successful in classifying difficult and simple queries by quantifying this distinction. For moderate queries, on the other hand, finding the distinction has not been easy for SQD_{PGB}.

Table 3 depict that SQD_{PGB} has a more correlated performance to the actual values than SQD_{GB}, SQD_{CB}, and SQD_{ChB}. To analyze the reason, let us look at how semantic similarities are approximated in these different models more thoroughly. In ChB, semantic similarities between entity pairs are normalized by a coherence coefficient, and performance is very dependent on the



(a) ClueWeb09-B

(b) ClueWeb12-B

(c) Robust4

Fig. 2. Pearson correlation between simple, moderate, and hard queries' AP and estimated values by four semantic distinction (SD) predictors, and Weighted Information Gain (WIG) predictor on ClueWeb09-B, ClueWeb12-B, and Robust4. In the plot, the X-axis represents the queries' category and Y-axis represents Pearson correlation values.

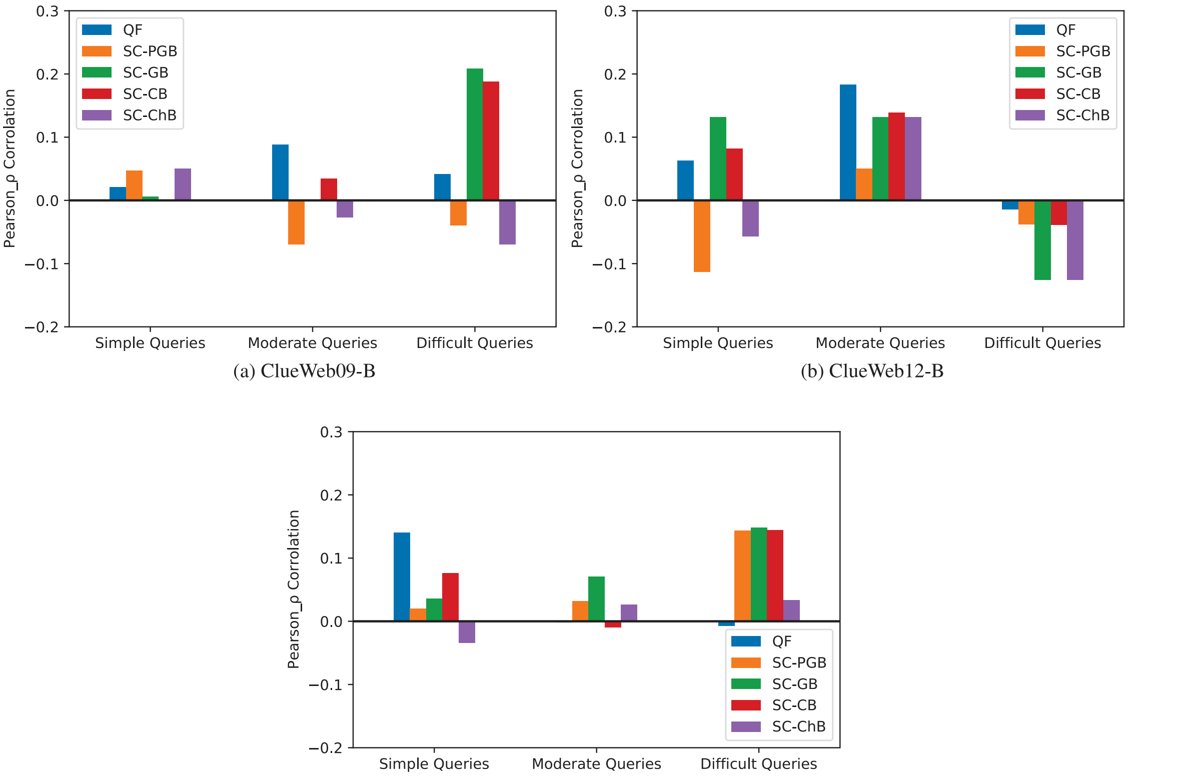
performance of the entity linking system. In GB and CB models, the best match between entities in a query and entities in a document is used for measuring their semantic similarities, and all other relatednesses between query and document entities are ignored. On the other hand, in the PGB similarity model, the similarity between all entity pairs from a given query and a document is exploited for calculating the semantic similarity score.

The following example shows why the PGB approach for calculating semantic similarity performs better for a query-drift predictor. Get the TREC topic 132 'Mother day song', linked to 'Mother's Day' entity by Tagme entity linking system as an example. With an AP of 0.0164, this query is among the difficult ones for the QL retrieval system over the ClueWeb09-B dataset. The retrieved documents include those with some unrelated entities referring to romantic songs (but not Mother's day songs). For SQD, the algorithm measures how close a top-ranked document is to the query compared to the retrieved documents' unrelated aspects.

Contrary to PGB, in both CB and GB similarity models, a document with many unrelated entities can be considered distinguishably closer to the query than the average if it has only one mention of the query's entity. In this example, if a document's primary focus and theme are not about Mother's day song and only contains one mention of 'mother's day', it is considered closer to query. The reason is all unrelated entities will be masked by a single occurrence of the "Mother's day" entity in the document. Observably, PGB is more effective than other similarity models in associating different scores to documents with varying levels of similarity and distinguishing between a highly related document and a document whose content is close to the query drift. That is what happened in this particular example, making SQD_{GB}, and SQD_{CB} categorizes this query as a simple one. In contrast, SQD_{PGB} correctly classifies it as a difficult one, i.e., the difference between each top-ranked document and the average regarding this query measured low correctly.

4.3.2. Semantic distinction (SD) predictors

Similarly to Section 4.3.1, we analyze the performance of SD predictors for simple, moderate and difficult queries by reporting the Pearson correlation coefficient ρ between the predicted performance by SD predictors in each category and the average precision for the top 1000 documents retrieved per query (AP@1000) based on the collection relevance judgements over ClueWeb09-B, ClueWeb12-B, and Robust04 datasets (Fig. 2). Fig. 2 also includes the Pearson correlation coefficient achieved by WIG, the prominent term-based predictor based on the query distinction hypothesis. This figure shows that the term-based predictor perform mostly weaker than semantic-based predictors for simple and moderate queries.



(a) ClueWeb09-B

(b) ClueWeb12-B

(c) Robust4

Fig. 3. Pearson correlation between simple, moderate, and hard queries' AP and estimated values by four semantic cohesion predictors (SC) predictors, and Query Feedback (QF) predictor on ClueWeb09-B, ClueWeb12-B, and Robust4. In the plot, the X-axis represents the queries' category and Y-axis represents Pearson correlation values.

That confirms our discussion in Section 3.1 that semantic methods have stronger capabilities for measuring similarities between a set of retrieved documents and the query. These figures also show that all distinction-based predictors, including the baseline term-based and the introduced semantic-based one, struggle to predict difficult queries.

Contrary to the SQD predictors, the semantic similarity model PGB has no clear advantage over the other semantic predictors in SD predictors. The reason can be due to the different ways SD and SQD approach query performance prediction. In SQD-based algorithms, it is essential to measure a retrieved document's closeness to unrelated aspects (represented by the query drift). While in SD, the relatedness of a document to the query is measured, and the unrelated aspects of top-ranked documents have no role in predicting query performance. Hence, both SD_{GB} and SD_{CB} models, which rely on finding the best matches for the query entities inside the set of document entities and use them for measuring the relatedness of the query and the document, work well in approximating semantic similarities.

4.3.3. Semantic cohesion (SC) predictors

Fig. 3 reports the Pearson correlation coefficient ρ between the predicted performance by the semantic cohesion predictors (SC) in each category and the average precision for the top 1000 documents retrieved per query (AP@1000) based on the collection relevance judgements over ClueWeb09-B, ClueWeb12-B, and Robust04 datasets. This figure also reports the Pearson correlation coefficient achieved by QF, a prominent term-based predictor that attempts to measure the coherence of top-ranked documents as a sign of query performance. There is a notable pattern in the ClueWeb12-B dataset (Fig. 3), where all predictors, including the term-based one, have negative correlation values on difficult queries. SC_{PGB} and SC_{ChB} retain this pattern and have negative correlations on simple queries as well. This means that the cohesion between the retrieved documents' content is not a clear sign for query performance over the ClueWeb12-B dataset. This justifies the findings in recent QPP works, including this work, for interpolating different predictors for achieving a stable prediction performance over different datasets. Table 3 shows that SC_{ChB} is mostly performing weaker than the others on different datasets. Contrary to SD and SQD, which exploit similarities between a query (usually a short line of text) to documents, SC needs to compare the similarity between two documents (usually two long texts). This observation concludes that ChB is weaker than other similarity methods compared to the similarity of two long pieces of text for QPP.

Table 4
Statistics on single-entity and complex queries.

Collection	#Single-Entity Queries	#Complex Queries
ClueWeb09-B	77	115
ClueWeb12-B	13	35
Robust04	43	181

4.4. Prediction performance for single-entity queries

Queries that ask about a single entity form an important category that has been the main focus of a growing body of research in question answering, entity ranking, entity retrieval, and knowledge graph search in recent years (da Silva et al., 2020; Huang, Zhang, Li, & Li, 2019; Lukovnikov, Fischer, & Lehmann, 2019). TREC Web Track Query #201, 'raspberry pi', which is linked to the Wikipedia page #5265384 with the same name **rasberry Pi**, Query #202, 'uss carl vinson', which is linked to the Wikipedia page #197352 with the same name **USS Carl Vinson**, and Query #247, 'rain man', which is linked to the Wikipedia page #129368 with the same **Rain Man** are samples of single-entity queries. In this section, we empirically analyze how effective the semantic predictors are in predicting single-entity queries' performance.

For evaluation, we provide a classification of TREC queries publicly available.[5] In this classification, all TREC queries are classified into two classes:

- The single-entity queries ('raspberry pi' (#201), 'rain man' (#201), 'yahoo' (#23), and 'neil young'(#73)).
- complex queries, queries that are either linked to more than one entities (e.g., TREC Web Track Query #11, 'gmat prep classes', which is linked to two entities: **Test preparation**, Wikipedia page #23197648, and **Graduate Management Admission Test** Wikipedia page #255232) or are linked to one entity by our entity linking system, but request more information than the entity itself (e.g., TREC Web Track Query #110, 'map of brazil', which is linked to the Wikipedia page #3383, **Brazil** but requests more information. This means that those documents that have general information about Brazil can be unrelated to this query).

Table 4 shows the statistics on single-entity and complex queries. Table 5 shows the Kendall correlation coefficient of the semantic predictors and term-based baselines on these two categories of queries, on Robust04, ClueWeb09-B, ClueWeb12-B datasets. For ClueWeb12-B, most baseline predictors achieve a very low Kendall correlation coefficient for single-entity queries. SCM_{PGB} and SCM_{GB} are the strongest predictors among semantic predictors (with Kendall τ of 0.271 and 0.194, respectively). Here, the noteworthy fact is that contrary to most predictors, SCM_{GB} achieves even better performance over single-entity queries compared with complex queries. For Robust04 dataset, SCM_{CB}, SCM_{GB}, and SCM_{PGB} outperform all baseline predictors for single-entity queries. The notable observation in Robust04 is that semantic predictors are much stronger than baselines in predicting the performance of single-entity queries but show close or even worse performance over complex queries. The only exception is SCM_{GB} that outperforms all baselines in both single-entity and complex queries. Finally, for ClueWeb09-B, SCM_{CB} and SCM_{GB} are the strongest predictors that outperform all baselines on single-entity queries. For complex queries, SCM_{CB} and SCM_{ChB} (with Kendall τ of 0.225 and 0.229, respectively) are the strongest predictors comparing to all baselines.

4.5. Discussion

In Section 4, we defined three research objectives to analyze whether the proposed predictors are effective in three different situations. First, to predict the performance of queries across different datasets (Research Objective 1), second, to investigate the impact of semantic similarity models on the proposed predictors' performance (Research Objective 2); and third, to analyze the proposed predictors' effectiveness for single-entity queries (Research Objective 3). This section analyzes these research questions based on the results reported in Sections Section 4.2, 4.3, and 4.4.

Regarding the first research objective, the following conclusions can be derived from the experiments:

- Several of our proposed predictors are more effective than the state-of-the-art methods for predicting the performance of queries over two Web-scale datasets of ClueWeb09-B and ClueWeb12-B and are among the most effective ones over the Robust094 dataset.
- The linear interpolation of the proposed semantic predictors contributes to a more stable prediction performance over different datasets.
- The linear interpolation of the proposed semantic predictors with term-based predictors (such as WIG, Clarity, and $UEF_{Clarity}$) leads to improved performance over the baselines.

Table 6 reports the introduced predictors that outperform all baselines or are among the top-three ones across different datasets. These results are important because the baselines either have not reported the results on all of these databases or report a poor performance on one while reaching a good performance over the other datasets.

[5] https://github.com/ParastooSJ/SemanticQPP.

Table 5
Kendall correlation coefficient τ for the correlation between the predicted performance by QPP methods and the average precision for the top 1000 documents retrieved per query (AP@1000) based on the collection relevance judgments for Single-entity Queries, denoted as SQuery, and Complex Queries, denoted as CQuery. Bold numbers show the proposed semantic combination metrics values that are higher than all baselines in each column.

Method		Robust04		ClueWeb09-B		ClueWeb12-B	
		SQuery	CQuery	SQuery	CQuery	SQuery	CQuery
Baselines	Clarity	−0.116	−0.007	0.247	0.169	−0.039	0.274
	NQC	0.357	0.330	0.168	0.155	−0.013	−0.038
	WIG	−0.110	0.074	0.181	0.086	−0.039	0.038
	SMV	0.272	0.297	0.214	0.161	0.039	0.264
	QF	0.278	0.313	0.050	0.014	0.199	−0.022
	UEF$_{Clarity}$	0.151	0.171	0.252	0.174	−0.090	0.271
	UEF$_{WIG}$	0.176	0.242	0.160	0.090	0.090	0.137
PGB	SQD$_{PGB}$	0.326*	0.176	0.175	0.119*	0.400	0.096
	SD$_{PGB}$	0.556*	0.310*	0.253*	0.160*	0.219	0.228
	SC$_{PGB}$	0.384*	0.260*	−0.080	−0.126	0.142	−0.234
	SCM$_{PGB}$	**0.493***	0.305*	**0.254***	**0.184***	**0.271**	**0.363**
GB	SQD$_{GB}$	0.112*	0.057*	−0.041	−0.001	0.234	−0.051
	SD$_{GB}$	0.371*	0.296	0.233*	0.211	0.219	0.298
	SC$_{GB}$	0.484*	0.316*	0.308*	0.168*	0.013	0.049
	SCM$_{GB}$	**0.519***	**0.345***	**0.311***	**0.186***	0.194	0.019
CB	SQD$_{CB}$	−0.229	−0.129	−0.081	−0.101	−0.104	−0.054
	SD$_{CB}$	0.462*	0.208*	0.286*	0.254*	0.219	0.336
	SC$_{CB}$	0.522*	0.284*	0.302	0.186*	0.013	0.120
	SCM$_{CB}$	**0.508***	0.257*	**0.312***	**0.225***	0.116	0.238
ChB	SQD$_{ChB}$	0.044*	0.068*	−0.093	−0.071	0.182	−0.122
	SD$_{ChB}$	0.311*	0.252*	0.289	0.222*	0.348	0.238
	SC$_{ChB}$	0.125	0.173*	−0.013	−0.102	0.168	−0.373
	SCM$_{ChB}$	0.247*	0.232	0.235*	**0.229***	0.142	**0.292**

Table 6
Semantics-based predictors that outperform all baselines (noted by ✓) or are among the top-three ones (noted by ✠). For more detailed information please see Section 4.2.

	ClueWeb09-B	ClueWeb12-B	Robust
SCM$_{PGB}$	✠	✓	✠
SCM$_{PGB}$+Clarity	✓	✓	✠
SCM$_{PGB}$+UEF$_{Clarity}$	✓	✓	✠
SCM$_{PGB}$+WIG	✠	✓	✠

Regarding the second research objective, our experiments show that the choice of semantic similarity model impacts the performance of the proposed predictors. Sim_{PGB} is the most effective semantic similarity model when employed in the SQD predictor, and Sim_{ChB} is the weakest semantic similarity model when employed in the SC predictor. The findings for the second research objective can be summarized as follows:

- Sim$_{PGB}$ is the most effective semantic similarity method when employed by the SQD predictor, especially in simple and difficult queries.
- No semantic similarity method has a clear advantage over the others when employed by the SD predictors, and all struggle in predicting difficult queries.
- Sim$_{ChB}$ is the weakest semantic similarity measure employed by SC, especially for predicting simple and difficult queries.

Table 7 summarize this discussion by showing the best semantic similarity models for the proposed predictors. As we can see in this table, while ChB is not a recommended similarity model, PGB can be employed for SQD and SCM, GB can be used for SD and SC, and CB is good choice for SD and SCM. Using similarity models recommended by our experiments improve the prediction performance.

Finally, in terms of the third research objective, our experiments illustrate that the proposed predictors outperform all baselines on simple questions, i.e., queries that ask about a single entity in the knowledge graph. The following main findings are derived from the experiments depicted in experimental results:

- The SCM predictors, created based on a linear interpolation of the proposed predictors, are much stronger than baselines in predicting single-entity queries' performance.
- Similar to the findings for the first research objective, we can observe that the proposed predictors, without interpolation, show highly divergent performance over different datasets for single-entity queries.

Table 7
Best semantic similarity models for the proposed semantic predictors.

	PGB	GB	CB	ChB
SQD	✓			
SD		✓	✓	
SC		✓		
SCM	✓		✓	

5. Concluding remarks

This paper presented a semantics-based approach for post-retrieval query-performance prediction based on measuring semantic similarities between entities in documents and queries. We introduced predictors for measuring semantic distinction, semantic query drift, and semantic cohesion in the top-ranked list of retrieved documents as indicators for measuring query performance. Our finding is that the introduced semantics approach is more effective in predicting query performance than the existing term-based methods by considering semantic relatedness instead of exact term-match.

We empirically evaluated the proposed predictors' effectiveness through a complete set of experiments on Robust04, ClueWeb09-B, and ClueWeb12-B datasets. We compared the ranking of queries based on the proposed predictors' prediction and the real ranking based on the relevance judgment. We reported Pearson and Kendall Correlation rank coefficients as the correlation between predictions and actual data.

The standard approach in QPP literature is to predict the query likelihood model's performance, which is a term-based baseline retrieval system. Our work shows that the employment of semantic predictors improves the prediction performance of the existing methods. Given a rich body of work in employing knowledge graph entities in document search (Dietz et al., 2018) and semantic search systems, we believe there is room for developing and analyzing QPP methods on the foundations of our work that go beyond predicting the query likelihood's performance model.

References

Al-Khatib, K., Hou, Y., Wachsmuth, H., Jochim, C., Bonin, F., & Stein, B. (2020). End-to-end argumentation knowledge graph construction. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34*, (pp. 7367–7374).

AlMousa, M., Benlamri, R., & Khoury, R. (2021). Exploiting non-taxonomic relations for measuring semantic similarity and relatedness in WordNet. *Knowledge-Based Systems*, *212*, Article 106565.

Amati, G., Carpineto, C., & Romano, G. (2004). Query difficulty, robustness, and selective application of query expansion. In *Proceedings of the European conference on information retrieval* (pp. 127–137). Springer.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web* (pp. 722–735). Springer.

Ayvaz, S., & Aydar, M. (2019). Dynamic discovery of type classes and relations in semantic web data. *Journal on Data Semantics*, *8*(1), 57–75.

Azad, H. K., & Deepak, A. (2019). Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, *56*(5), 1698–1735.

Bagheri, E., Arabzadeh, N., Zarrinkalam, F., Jovanovic, J., & Al-Obeidat, F. (2020). Neural embedding-based specificity metrics for pre-retrieval query performance prediction. *Information Processing & Management*, *57*(4), Article 102248.

Białecki, A., Muir, R., Ingersoll, G., & Imagination, L. (2012). Apache lucene 4. In *SIGIR 2012 workshop on open source information retrieval* (p. 17).

Callan, J., Hoy, M., Yoo, C., & Zhao, L. (2009). The clueweb09 dataset, 2009. URL http://boston.lti.cs.cmu.edu/data/clueweb09.

Carmel, D., & Yom-Tov, E. (2010). Estimating the query difficulty for information retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval* (p. 911).

Chifu, A.-G., Laporte, L., Mothe, J., & Ullah, M. Z. (2018). Query performance prediction focused on summarized letor features. In *Proceedings of the 41th international ACM SIGIR conference on research & development in information retrieval* (pp. 1177–1180).

Cormack, G. V., Smucker, M. D., & Clarke, C. L. (2011). Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, *14*(5), 441–465.

Cornolti, M., Ferragina, P., & Ciaramita, M. (2013). A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on world wide web* (pp. 249–260).

Cover, T. M., & Thomas, J. A. (1991). Entropy, relative entropy and mutual information. *Elements of Information Theory*, *2*, 1–55.

Croft, W. B., Metzler, D., & Strohman, T. (2010). *Search engines: Information retrieval in practice, Vol. 520*. Addison-Wesley Reading.

Cronen-Townsend, S., Zhou, Y., & Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 299–306).

da Silva, J. W. F., Venceslau, A. D. P., Sales, J. E., Maia, J. G. R., Pinheiro, V. C. M., & Vidal, V. M. P. (2020). A short survey on end-to-end simple question answering systems. *Artificial Intelligence Review*, *53*(7), 5429–5453.

Déjean, S., Ionescu, R. T., Mothe, J., & Ullah, M. Z. (2020). Forward and backward feature selection for query performance prediction. In Proceedings of the 35th annual ACM symposium on applied computing, (pp. 690–697).

Diaz, F. (2007). Performance prediction using spatial autocorrelation. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 583–590).

Dietz, L., Kotov, A., & Meij, E. (2018). Utilizing knowledge graphs for text-centric information retrieval. In *Proceedings of the 41st international ACM SIGIR conference on research & development in information retrieval* (pp. 1387–1390).

Ensan, F., & Bagheri, E. (2017). Document retrieval model through semantic linking. In *Proceedings of the tenth ACM WSDM international conference on web search and data mining* (pp. 181–190).

Fafalios, P., Kasturia, V., & Nejdl, W. (2018). Ranking archived documents for structured queries on semantic layers. In *Proceedings of the 18th ACM/IEEE on joint conference on digital libraries* (pp. 155–164).

Feng, Y., Bagheri, E., Ensan, F., & Jovanovic, J. (2017). The state of the art in semantic relatedness: a framework for comparison. *The Knowledge Engineering Review*, *32*, Article e10.

Ferragina, P., & Scaiella, U. (2010). Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM CIKM international conference on information and knowledge management* (pp. 1625–1628).

Gabrilovich, E., Markovitch, S., et al. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on artificial intelligence, vol. 7* (pp. 1606–1611).

Hauff, C. (2010). Predicting the effectiveness of queries and retrieval systems. *SIGIR Forum*, *44*(1), 88.

He, B., & Ounis, I. (2004). Inferring query performance using pre-retrieval predictors. In *International symposium on string processing and information retrieval* (pp. 43–54). Springer.

He, B., & Ounis, I. (2006). Query performance prediction. *Information Systems*, *31*(7), 585–594.

He ClueWeb12 dataset: Dataset details. (2021). https://lemurproject.org/clueweb12/specs.php. (Accessed: 2021-03-26).

Huang, X., Zhang, J., Li, D., & Li, P. (2019). Knowledge graph embedding based question answering. In *Proceedings of the twelfth ACM international conference on web search and data mining* (pp. 105–113).

Hulpuş, I., Prangnawarat, N., & Hayes, C. (2015). Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In *Proceedings of the 14th international semantic web conference* (pp. 442–457). Springer.

Hussain, M. J., Wasti, S. H., Huang, G., Wei, L., Jiang, Y., & Tang, Y. (2020). An approach for measuring semantic similarity between Wikipedia concepts using multiple inheritances. *Information Processing & Management*, *57*(3), Article 102188.

Karisani, P., Rahgozar, M., & Oroumchian, F. (2016). A query term re-weighting approach using document similarity. *Information Processing & Management*, *52*(3), 478–489.

Kelbert, M., Stuhl, I., & Suhov, Y. (2017). Weighted entropy: basic inequalities. *Modern Stochastics: Theory and Applications*, *4*(3), 233–252.

Krikon, E., Carmel, D., & Kurland, O. (2012). Predicting the performance of passage retrieval for question answering. In *Proceedings of the 21st ACM CIKM international conference on information and knowledge management* (pp. 2451–2454).

Kurland, O., Shtok, A., Carmel, D., & Hummel, S. (2011). A unified framework for post-retrieval query-performance prediction. In *Proceedings of the third international conference on advances in information retrieval theory* (pp. 15–26).

Laclavík, M., & Ciglan, M. (2013). Towards entity search: Research roadmap. *Proceedings of WIKT*, 161–166.

Lashkari, F., Bagheri, E., & Ghorbani, A. A. (2019). Neural embedding-based indices for semantic search. *Information Processing & Management*, *56*(3), 733–755.

Li, P., Xiao, B., Ma, W., Jiang, Y., & Zhang, Z. (2017). A graph-based semantic relatedness assessment method combining wikipedia features. *Engineering Applications of Artificial Intelligence*, *65*, 268–281.

Liu, Y., Hua, W., & Zhou, X. (2021). Temporal knowledge extraction from large-scale text corpus. *World Wide Web*, *24*(1), 135–156.

Lukovnikov, D., Fischer, A., & Lehmann, J. (2019). Pretrained transformers for simple question answering over knowledge graphs. In *Proceedings of the 18th international semantic web conference* (pp. 470–486). Springer.

Ma, H., Alipourlangouri, M., Wu, Y., Chiang, F., & Pi, J. (2019). Ontology-based entity matching in attributed graphs. *Proceedings of the VLDB Endowment*, *12*(10), 1195–1207.

Mendes, P. N., Jakob, M., García-Silva, A., & Bizer, C. (2011). DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems* (pp. 1–8).

Moro, A., Raganato, A., & Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, *2*, 231–244.

Ni, Y., Xu, Q. K., Cao, F., Mass, Y., Sheinwald, D., Zhu, H. J., et al. (2016). Semantic documents relatedness using concept graph representation. In *Proceedings of the ninth ACM international conference on web search and data mining* (pp. 635–644).

Nunes, B. P., Dietze, S., Casanova, M. A., Kawase, R., Fetahu, B., & Nejdl, W. (2013). Combining a co-occurrence-based and a semantic measure for entity linking. In *Extended semantic web conference* (pp. 548–562). Springer.

Nunes, B. P., Kawase, R., Fetahu, B., Dietze, S., Casanova, M. A., & Maynard, D. (2013). Interlinking documents based on semantic graphs. *Procedia Computer Science*, *22*, 231–240.

Paul, C., Rettinger, A., Mogadala, A., Knoblock, C. A., & Szekely, P. (2016). Efficient graph-based document similarity. In *Proceedings of European semantic web conference* (pp. 334–349). Springer.

Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 275–281).

Pound, J., Mika, P., & Zaragoza, H. (2010) Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th international conference on world wide web* (pp. 771–780).

Qu, R., Fang, Y., Bai, W., & Jiang, Y. (2018). Computing semantic similarity based on novel models of semantic representation using Wikipedia. *Information Processing & Management*, *54*(6), 1002–1021.

Raiber, F., & Kurland, O. (2014). Query-performance prediction: setting the expectations straight. In *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval* (pp. 13–22).

Raviv, H., Kurland, O., & Carmel, D. (2014). Query performance prediction for entity retrieval. In *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval*, (pp. 1099–1102).

Roitman, H. (2017). An enhanced approach to query performance prediction using reference lists. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 869–872).

Roitman, H. (2019a). Normalized query commitment revisited. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 1085–1088).

Roitman, H. (2019b). Query term weighting based on query performance prediction. arXiv preprint arXiv:1902.10371.

Roy, D., Ganguly, D., Mitra, M., & Jones, G. J. (2019). Estimating gaussian mixture models in the local neighbourhood of embedded word vectors for query performance prediction. *Information Processing & Management*, *56*(3), 1026–1045.

Shtok, A., Kurland, O., & Carmel, D. (2016). Query performance prediction using reference lists. *ACM Transactions on Information Systems (TOIS)*, *34*(4), 19.

Shtok, A., Kurland, O., Carmel, D., Raiber, F., & Markovits, G. (2012). Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems (TOIS)*, *30*(2), 1–35.

Strube, M., & Ponzetto, S. P. (2006). WikiRelate! Computing semantic relatedness using Wikipedia. In *Proceedings of the 21st national conference on artificial intelligence vol. 6* (pp. 1419–1424).

Tao, Y., & Wu, S. (2014). Query performance prediction by considering score magnitude and variance together. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management* (pp. 1891–1894).

Traverso, I., Vidal, M.-E., Kämpgen, B., & Sure-Vetter, Y. (2016). GADES: A graph-based semantic similarity measure. In *Proceedings of the 12th International Conference on Semantic Systems* (pp. 101–104).

Voorhees, E. M., et al. (2005). Overview of the TREC 2005 robust retrieval track. In *Trec*.

Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y., et al. (2020). Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 23–30). Association for Computational Linguistics.

Zamani, H., Croft, W. B., & Culpepper, J. S. (2018). Neural query performance prediction using weak supervision from multiple signals. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 105–114).

Zhao, Y., Scholer, F., & Tsegay, Y. (2008). Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Proceedings of the IR research, 30th european conference on advances in information retrieval* (pp. 52–64). Springer.

Zhao, C., Xiong, C., Qian, X., & Boyd-Graber, J. (2020). Complex factoid question answering with a free-text knowledge graph. In *Proceedings of the web conference 2020* (pp. 1205–1216).

Zhou, Y., & Croft, W. B. (2006). Ranking robustness: a novel framework to predict query performance. In *Proceedings of the 15th ACM CIKM international conference on information and knowledge management* (pp. 567–574).

Zhou, Y., & Croft, W. B. (2007). Query performance prediction in web search environments. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 543–550).

Zou, J., Chen, Y., & Kanoulas, E. (2020). Towards question-based recommender systems. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 881–890).