Semantics-enabled Query Performance Prediction for Ad hoc Table Retrieval

Maryam Khodabakhsh[a,1], Ebrahim Bagheri[b]

[a] *Computer Engineering Department, Shahrood University of Technology, Shahrood, Iran*
[b] *Laboratory for Systems, Software and Semantics (LS^3), Ryerson University*

Abstract

Predicting the performance of a retrieval method for a given query is a highly important and challenging problem in information retrieval. Accurate Query Performance Prediction (QPP) plays an important role in real time handling of queries with varying levels of difficulty. While there have been several successful query performance predictors, no predictors have yet been introduced within the context of the ad hoc table retrieval task, which is concerned with answering a query with a ranked list of tables. In this paper, we propose to perform query performance prediction based on neural embedding techniques for ad hoc table retrieval and introduce three neural features. The neural features are based on neural embedding techniques and leverage the distance between tokens in the embedding space in order to capture their semantic similarity. We evaluate our proposed work based on a gold standard test collection and compare it with the state-of-the-art post-retrieval query performance prediction methods. We find that our neural features (1) are effective for predicting the performance of content-based ranking functions; and not as effective for feature-based ranking functions, and (2) show a synergistic impact on existing QPP methods and hence are able to increase their performance in practice.

Keywords: Query Performance Prediction, Ad hoc Table Retrieval, Neural Embeddings

1. Introduction

Within the context of ad hoc retrieval, the effectiveness of information retrieval (IR) methods can significantly vary across different sets of queries and hence the quality of the results returned for

*Corresponding Author
Email addresses: m_khodabakhsh@shahroodut.ac.ir (Maryam Khodabakhsh), bagheri@ryerson.ca (Ebrahim Bagheri)

some queries can be poorer compared to others. Such queries are often known as 'difficult' queries. Thus, early and real-time determination of those queries that are more difficult than others can help IR systems choose appropriate strategies for dealing with them through techniques such as query reformulation [1].

Query Performance Prediction (QPP) methods are specifically designed to identify the difficulty of a query by estimating the performance of a retrieval system for that given query in the absence of relevance judgments. Query performance prediction has been studied well in the past few years most specifically for the task of ad hoc document retrieval [2, 3, 4]. Empirical studies have shown that the quality of existing predictors can still be improved specially when considering factors such as query ambiguity, missing content, and vocabulary mismatch [5], which are often hard to address because most predictor models are based on features such as the frequency of query terms [6] or the average IDF of query terms in the collection [7]. These features are primarily frequency-based term-dependent features and cannot address issues such as vocabulary mismatch.

While existing work in query performance prediction has been predominantly focused on various statistical measures based on term frequency and lexical matching, the information retrieval community has recently embarked on exploring the impact and importance of neural information retrieval techniques [8, 9, 10] or latent models [11, 12] where a document that does not overlap in terminology with the query can still be retrieved. The benefit of such methods is that they consider semantic association between the document and query spaces and hence are able to bridge the gap between these two spaces when the vocabulary mismatch problem is observed [8, 9, 11, 12]. Empirical experimentation has shown that such features are strong complementary features for term-based features and can lead to noticeable improvement in ad hoc retrieval [13]. While such features have been widely adopted in the recent years for performing retrieval, they have only been investigated in the context of ad hoc *document* retrieval or QPP for ad hoc *document* retrieval.

In this paper, we are interested specifically in query performance prediction in ad hoc *table* retrieval [14, 15]. The task of ad hoc table retrieval is concerned with the identification of the most relevant full tables to a given input query. The effective retrieval of relevant tables is important as it facilitates the identification of information that are structured in tabular form. These include tabular data from the financial sector, demographic information, catalogues and even database relational tables, just to name a few. As argued by Zhang and Balog [14], the effective retrieval

of tables has strong impact on other dependent tasks such as table completion and table mining. While ad hoc table retrieval is a subset of the ad hoc retrieval task, it can be considered to be more challenging than ad hoc document retrieval, as unlike documents, data in a table lack continuity and context that would be helpful for measuring relevance. Therefore, ad hoc table retrieval has been gaining increasing attention from the research community in the recent years [15, 16, 17, 14, 18, 19].

Recent work in ad hoc table retrieval [15] has shown that various retrieval methods can have complementary behavior. In other words, the best table retrieval methods in the literature do not necessarily perform better on all range of queries. Therefore, it is essential to be able to predict which method has the best performance on any given input query to be able to effectively route the query to the best retrieval method. As such, our work in this paper is focused on query performance prediction in the context of ad hoc table retrieval, which has not been explored in the literature. We will investigate how existing post-retrieval query performance predictors perform in the context of ad hoc table retrieval and subsequently systematically propose to integrate semantic neural information into performance prediction for building stronger predictors.

To this end, our work rests on and explores a fundamental idea that given the user's query and a retrieved document set, features that are based on some notion of frequency would not be sufficient features for predicting the performance of the retrieval method. It is our hypothesis that performance is appropriately estimated if semantic features are also taken into consideration. For this purpose, we predict query performance based on well-known neural embedding techniques [20, 21, 22] to measure the neural characteristics of the retrieved document set and the query.

Our proposed neural predictors have two main components: 1) in the first component, we define a representation function to transform the raw content of queries and tables into a bag of tokens (tokens can be keywords or entities), and 2) in the second component, various neural features based on neural embedding techniques are proposed and integrated into the state-of-the-art post-retrieval predictor models. The key contributions of our work can be summarized as follows:

1. We systematically investigate the impact of the state-of-the-art post-retrieval query performance prediction methods in the context of ad hoc table retrieval;

2. We show how neural features derived from neural embedding techniques can be integrated into existing query performance predictors, leading to neural predictors;

3. We evaluate our proposed neural predictors on a gold standard test collection derived from Wikipedia tables. We further analyze our findings from various perspectives and identify areas where neural predictors are able to provide effective performance improvements.

In our experiments, we will explore and systematically answer four research questions, including (1) whether the representation of table content in the form of keywords or entities have an impact on determining query difficulty; (2) would it be possible to improve the estimation power of existing post-retrieval query performance predictors using neural predictors that are proposed in this paper; (3) Would the proposed neural predictors have a complementary impact on existing QPP methods; and (4) Whether the consideration of table specific structural characteristics such as table cells, rows and columns have any impact on the performance of our proposed neural predictors.

The rest of this paper is organized as follows. In the next section, we review the related work in two subsections covering pertinent work in ad hoc table retrieval and query performance prediction techniques. Section 3 provides the technical details of our proposed neural predictors. The details of our experiments, gold standard test collection, baselines and our findings are presented in Section 4. Finally, the paper is concluded in Section 5 with a summary of our findings.

2. Related Work

In this section, we first review prior work on ad hoc table retrieval, and then present the related literature on query performance prediction.

2.1. Ad Hoc Table Retrieval

Tables are suitable tools for grouping large volumes of information in one simplified structure. They are widely adopted for reporting data on the Web and have found a special place on Wikipedia. Various researchers have already explored various aspects of table analytics such as mining [23], searching [14, 24, 25], extracting [24], completing [26] and generating [27] tables in Web documents. The focus of our paper is on the task of searching for and retrieving tables in response to a given unstructured query, often known as ad hoc table retrieval. A full account of the relevant state of the art can be found in a very recent paper on this topic [18]. Existing work in ad hoc table retrieval can be categorized from two perspectives: (1) the types of the *ranking function* used for

ranking tables, and (2) the *features* used in the retrieval methods. We review the related literature from these two views:

Ranking Functions. The objective of a ranking function is to rank a set of documents based on their relevance to the query, which can be in the form of feature-based learn to rank methods [28, 29], statistical and probability techniques such as language models [30], entity linking-based techniques [31, 32], models that rely on diversity of table content [33], network properties [34] or knowledge graph-based similarity [35]. More recent techniques capture the semantic aspect of the tables and queries in the process of ranking. For instance, Zhang and Balog [14] were the first to develop a ranking method to match queries and tables based on their semantic association where both queries and tables were represented using semantic concepts (bag-of-entities and bag-of-categories) as well as continuous dense vectors (word and graph embeddings). Additional work was then subsequently developed to learn neural representations for tables for the purpose of retrieval [17].

Features. Ranking functions rely on a set of features extracted from the query and table spaces. Some of these features are *table-specific*, which means that they represent the specific properties of a table such as the number of empty table cells in the table or the number of times a table has been viewed in the past [23]. Another group of features are *table-query interdependent* features, which measure the association of query and table pairs, e.g., total query term frequency in the table body [24]. The third type of feature focuses on measuring *query-specific* characteristics such as the length of the query [36]. This third type of feature is often the least discriminative form of features for ranking.

2.2. Query Performance Prediction

Query performance prediction models can be classified into pre-retrieval and post-retrieval models [37]. Pre-retrieval models predict the quality of the search results using only the raw query, and its descriptive features gathered at indexing time, e.g., the frequency of individual query terms in the target corpus [6]. Unlike pre-retrieval models where the predictions take place before the retrieval process, post-retrieval models analyze the search results of a query after retrieval in order to make a judgement about the difficulty of the query. Extensive research in the literature has already shown that post-retrieval methods for QPP are much stronger compared to pre-retrieval methods [38, 39]. As such, the focus of our work in this paper will be on post-retrieval models.

There are three broad categories of post-retrieval QPP models in the literature, namely Clarity-oriented models, Robustness-oriented models and Score-oriented models.

Clarity-oriented models [40] predict query performance by measuring the clarity of the returned results with respect to the corpus. The hypothesis of these models is that users expect the results to focus on the specific topics covered by the query; thus, the more related keywords to the query are observed in the result set, the less difficult the query would be. To this end, Clarity computes the difference between the likelihood of the most frequent keywords in the results and their likelihood in the whole corpus using measures such as the Kullback-Leibler divergence. If the language model of the results are similar to the language model of the whole corpus, then one can conclude that the query is difficult as it is not discriminatory enough.

Robustness-oriented models estimate query performance by measuring robustness with respect to perturbations in the query [2], in the documents [41], and in the retrieval method [42]. Intuitively, the more robust the result list is, the higher the performance of the query would be. One of the most popular robustness models is Query Feedback (QF) [2] that rebuilds a new query based on the result list. The correlation between the previous result list and the retrieved result list for the new query is an indication of robustness, which is then used to estimate query performance.

Measuring clarity and robustness is often a time-consuming process, which is due to the need to analyze the content of the result list. Less expensive alternatives include Score-oriented models, which use the score distribution of the retrieved documents. The basic idea behind these models is that the more a document is similar to a query, the higher the score of the document would be. Therefore, high scores in the top ranked documents are an indication of better retrieval performance and lower query difficulty. Weighted Information Gain (WIG) [2] and Normalized Query Commitment (NQC) [3] are two popular measures based on retrieval score. WIG essentially measures the divergence between the mean retrieval score of the top ranked documents and that of the whole corpus. It was shown that WIG is very effective for performing QPP on an MRF-based retrieval framework [43]. On the other hand, NQC measures the standard deviation of the retrieval scores normalized by the corpus score. Another Score-oriented model is Score Magnitude and Variance (SMV) [44] that considers both magnitude and variance of scores of the ranked list of results to measure the performance of a query.

In addition to the above three types of predictors, there are also other works, such as fusion

based retrieval [45], wherein several ranked lists, each one retrieved by a different ranking function, are combined into a single fused ranked list by a fusion method [46, 47]. NeuralQPP, from Zamani et al. [48], integrates three neural components, the component for top-k retrieval score, another for term distribution in the top retrieved documents and the last for representation of documents in the semantic space.

We would like to note that our work distinguishes itself from the existing literature as it (1) considers predicting query performance for ad hoc table retrieval, which has not been investigated in the past; and (2) it integrates information from neural embeddings and integrates it with existing QPP predictors, which is also a novel aspect of our work that has not been explored in the past.

3. Proposed Neural Query Performance Predictors

Let us begin by setting out the notation. Let q, C and R denote a query, a corpus, and a ranking function, respectively. A ranking function maps each table in C and query q into a real number, which represents the relevance degree. We use $\pi^k_R(q;C)$ to denote the list of the top k tables ranked by R in response to the query q. Our goal is to establish a neural predictor, NP, to quantify the effectiveness of $\pi^k_R(q;C)$ when no relevance judgments are available:

$$NP(\pi^k_R(q;C)) \stackrel{\text{def}}{=} P^{NF}_{base} \circ \varphi(\pi^k_R(q;C)) = P^{NF}_{base}(\varphi(\pi^k_R(q;C))) \tag{1}$$

$NP(\pi^k_R(q;C))$ is based on a function composition and quantifies the effectiveness of $\pi^k_R(q;C)$ based on some form of semantic information that can give way to inexact matching. The outer function, denoted as P^{NF}_{base}, can be any state-of-the-art post-retrieval query performance predictor. The predictor P^{NF}_{base} operates upon $\varphi(\pi^k_R(q;C))$ by employing semantic features. In our work, semantic features are implemented as *neural features* (NF) based on neural embedding techniques [21]. The inner function, denoted by φ, is a representation function that transforms query q and the tables in $\pi^k_R(q;C)$ to tokens such as keywords or entities. In the following subsections, we describe (1) the representation function φ, and (2) the connection between existing query performance predictors and neural features in more detail.

3.1. Representation function φ

In this subsection, we introduce the representation function φ to model the raw content of queries and the corpus as a bag of tokens where each token can be either a keyword or an entity.

In the context of ad hoc retrieval, corpus C is a collection of documents, where documents can be web pages [49], news articles [49], web tables [24], or images [50], among others. In this work, the documents are tables [51]. Also, query q represents the information need of the user.

Keyword-based representation. Given $\pi^k_R(q;C)$, for each table $T \in \pi^k_R(q;C)$, function φ takes the keywords in T and represents them as a bag of words (BOW).

Entity-based representation. In contrast to keyword-based representation, it is also possible to represent a table as a set of entities. Zhang and Balog show that some tables have an entity focus [52], which means that significant portions of the table have corresponding entity representations. Therefore, function φ uses entities of each table $T \in \pi^k_R(q;C)$ to represent the content of T as a bag of entities (BOE). The function φ can also transform query q to a bag of entities.

3.2. The Connection between QPP and Neural Features

As mentioned earlier, query performance predictors, such as WIG and Clarity, utilize term probability, which we denote as $p(\omega)$, where ω is considered in the context of a query, table or corpus in a strict term matching sense and disregard semantically related terms when computing term probability. As a result such predictors can suffer from not paying attention to the semantic association between terms, queries and tables. The objective of the outer function in Equation 1, i.e., P^{NF}_{base}, is to enable the inclusion of such semantic associations through measuring term similarities based on neural features.

In this paper, we introduce three main neural features (NFs) by using neural embedding techniques: (1) Neural Matching (NM), (2) Neural Aggregated Matching (NAM) and (3) Neural Distance (ND) and explain them in more detail in the following.

3.2.1. The Neural Matching (NM) Feature

One of the important applications of embedding techniques in IR is to employ the dense vector representations that they offer to model queries and documents (tables in our case) from the embeddings of the individual tokens that appear in each of them. This requires a systematic way to aggregate the embeddings of the individual tokens in the queries or documents (tables) to represent the whole query or document. Average Word (or token) Embedding (AWE) is a popular way for performing the aggregation [53, 54, 55]. Given table T and query q, AWE defines their

aggregated representations as follows:

$$\overrightarrow{\mathrm{T}} = \frac{1}{|\varphi(T)|} \sum_{t_T \in \varphi(T)} \frac{\overrightarrow{t_T}}{||\overrightarrow{t_T}||} \tag{2}$$

$$\overrightarrow{\mathrm{q}} = \frac{1}{|\varphi(q)|} \sum_{t_q \in \varphi(q)} \frac{\overrightarrow{t_q}}{||\overrightarrow{t_q}||} \tag{3}$$

where $\varphi(T)$ and $\varphi(q)$ are the output of the representation function φ that can be either bag of words or bag of entities for table T and query q, respectively. The table vector, denoted by $\overrightarrow{\mathrm{T}}$, is the centroid of all the normalized vectors for token t_T in the table and serves as a single embedding for the whole Table. Similarly, $\overrightarrow{\mathrm{q}}$ is the query vector.

Inspired by Equation 2, we define the corpus vector as $\overrightarrow{\mathrm{C}} = \frac{1}{|\varphi(C)|} \sum_{t_C \in \varphi(C)} \frac{\overrightarrow{t_C}}{||\overrightarrow{t_C}||}$ where $\varphi(C)$ is the tokens of corpus C denoted as $\varphi(C) = \cup_{T \in C} \varphi(T)$. In other words, $\varphi(C)$ is a bag of all of the table tokens.

In ad hoc table retrieval, the relevance of the retrieved tables to the input query is of foremost importance. Now, given that we have built vectors for both tables as well as queries, our objective is to measure the degree of relevance between them. Therefore, it is possible to directly calculate the similarity between a given table and the query vector representation. We assume table T is related to query q if the table vector is semantically similar to the query vector. Our proposed Neural Matching (NM) feature measures this semantic similarity using a vector similarity metric, such as cosine similarity or dot-product.

The intuition behind this feature is based on how tokens are adopted to express the content of a table. The assumption is that if the set of tokens representing a table is accurately selected, then any other query submitted from the users will use tokens that will be semantically close to the table representation in the embedding space. Therefore, given the fact that the position of a table can be determined based on the centroid of the position of its constituent tokens within the embedding space and also the position of a query can be determined in the same way in the same space, it is possible to measure the relevance of the tables to the queries using our proposed Neural Matching (NM) feature as follows:

Definition 1. ***(Neural Matching Feature)*** *Given $\overrightarrow{q}$ and $\overrightarrow{T}$, neural matching is equivalent to the cosine similarity between $\overrightarrow{q}$ and $\overrightarrow{T}$ and formally denoted as $NM(q, T)$.*

Table 1: The neural matching feature integrated into post-retrieval query performance predictors. T is a table in corpus C and $V(\varphi(C))$ is the vocabulary of the unique tokens in the corpus C. $\lambda(t_q)$ reflects the relative weight of the token's type t_q and is inversely related to the square root of the number of query tokens of that type. μ is the mean score of tables in $\pi^k_R(q;C)$. In the main Clarity formula [40], $\Pr\left(term \,\middle|\, \pi^k_R(q;C)\right)$ is the language model induced from the result list $\pi^k_R(q;C)$ and equals the sum over all tables in the result list. Here, it is replaced with $NM(t_C, \pi^k_R(q;C))$ in $P^{NM}_{Clarity}$. $NM(t_C, \pi^k_R(q;C) = \sum_{T\in\pi^k_R(q;C)} NM(t_C,T)(\prod_{t_q\in\varphi(q)} NM(t_q,T))$.

Base QPP Method	P^{NM}_{base}: Revised Post-retrieval Query Performance Predictors with Neural Matching Feature *(NM)*
Clarity [40]	$P^{NM}_{Clarity}(\varphi(\pi^k_R(q;C))) = \sum_{t_C\in V(\varphi(C))} NM(t_C, \pi^k_R(q;C)) \log \frac{NM(t_C,\pi^k_R(q;C))}{NM(t_C,C)}$
WIG [2]	$P^{NM}_{WIG}(\varphi(\pi^k_R(q;C))) = \frac{1}{k}\sum_{T\in\pi^k_R(q;C)} \sum_{t_q\in\varphi(q)} \lambda(t_q) \log \frac{NM(t_q,T)}{NM(t_q,C)}$
NQC [3]	$P^{NM}_{NQC}(\varphi(\pi^k_R(q;C))) = \frac{\sqrt{\frac{1}{k}\sum_{T\in\pi^k_R(q;C)}(NM(q,T)-\mu)^2}}{\lvert NM(q,C)\rvert}$
SMV [44]	$P^{NM}_{SMV}(\varphi(\pi^k_R(q;C))) = \frac{\sqrt{\frac{1}{k}\sum_{T\in\pi^k_R(q;C)}(NM(q,T)\lvert \ln \frac{NM(q,T)}{\mu}\rvert)}}{NM(q,C)}$

It is worth mentioning that the neural matching feature can be used for measuring the relevance of corpus C to query q given $\overrightarrow{\mathrm{q}}$ and $\overrightarrow{\mathrm{C}}$, denoted as $NM(q,C)$ or any two arbitrary vectors.

Now, given the neural matching feature, we are interested in using it in post-retrieval predictors in order to define P^{NF}_{base}. For this purpose, we adopt the widely used post-retrieval predictors reported in the literature as shown in Table 1 where NM has been integrated into the base post-retrieval predictors.

As shown in Table 1, Clarity [40] and WIG [2] require the probability of the query or corpus tokens in the tables or the corpus. Our proposed Neural Matching feature is an appropriate alternative for this probability as it measures the degree of relevance of each term to the table

Table 2: The neural aggregated matching feature integrated into post-retrieval query performance predictors. Table format is similar to Table 1. Also, $NAM(t_C, \pi^k_R(q;C) = \sum_{T \in \pi^k_R(q;C)} NAM(t_C, T)(\prod_{t_q \in \varphi(q)} NAM(t_q, T))$.

Base QPP Method	P^{NAM}_{base}: Revised Post-retrieval Query Performance Predictors with Neural Aggregated Matching Feature *(NAM)*
Clarity [40]	$P^{NAM}_{Clarity}(\varphi(\pi^k_R(q;C))) = \sum_{t_C \in V(\varphi(C))} NAM(t_C, \pi^k_R(q;C)) \log \frac{NAM(t_C, \pi^k_R(q;C))}{NAM(t_C, C)}$
WIG [2]	$P^{NAM}_{WIG}(\varphi(\pi^k_R(q;C))) = \frac{1}{k} \sum_{T \in \pi^k_R(q;C)} \sum_{t_q \in \varphi(q)} \lambda(t_q) \log \frac{NAM(t_q, T)}{NAM(t_q, C)}$
NQC [3]	$P^{NAM}_{NQC}(\varphi(\pi^k_R(q;C))) = \frac{\sqrt{\frac{1}{k} \sum_{T \in \pi^k_R(q;C)} (NAM(q,T) - \mu)^2}}{\lvert NAM(q,C) \rvert}$
SMV [44]	$P^{NAM}_{SMV}(\varphi(\pi^k_R(q;C))) = \frac{\sqrt{\frac{1}{k} \sum_{T \in \pi^k_R(q;C)} (NAM(q,T) \lvert \ln \frac{NAM(q,T)}{\mu} \rvert)}}{NAM(q,C)}$

or the corpus. Other predictors based on standard deviation, such as NQC [3] and SMV [44], employ a Score-oriented approach $Score(q, T)$, which indicates the score of table T for query q. By considering the fact that if a table achieves high relevance score for a given query, then the keywords or entities of the table will be semantically close to the keywords or entities used in the query in the embedding space, we revise these predictors by using $NM(q, T)$ instead of $Score(q, T)$.

It is worth noting that the vectors of the tables and the corpus can be calculated offline. Therefore, we will only need to measure the neural matching feature at runtime.

3.2.2. Neural Aggregated Matching (NAM) Feature

In this feature, we model each table as a bag of token vectors where vectors of tokens incorporate both syntactic and semantic aspects of the table content and therefore show to be effective for measuring the relevance degree. We adopt the same consideration for queries and the corpus and model them as a bag of token vectors. This is unlike the proposed NM feature where the queries,

tables and the corpus are represented by a vector in embedding space. Now, based on the bag of token vectors, we formally define the Neural Aggregated Matching (NAM) feature as follows:

Definition 2. ***(Neural Aggregated Matching Feature)*** *Given $\varphi(q)$ and $\varphi(T)$, neural aggregated matching is the maximum cosine of the token vector pairs in the $\varphi(q)$ and $\varphi(T)$ denoted as $NAM(q,T)$:*

$$NAM(q,T) = \max_{t_q \in \varphi(q), t_T \in \varphi(T)} (\{cos(\overrightarrow{t_q}, \overrightarrow{t_T})\}) \tag{4}$$

In this feature, the pairwise similarities between all table and query token vectors are computed and aggregated by a maximum function, which is a common approach to predict similarity [56]. Our assumption is that the maximum similarity captures the minimum cost required to match the query to the table.

The adopted formulation of the post-retrieval predictors based on NAM are provided in Table 2 where NAM is placed in the base post-retrieval predictors. This feature can determine the relevance between a query q and corpus C by giving $\varphi(q)$ and $\varphi(C)$, as $NAM(q,C)$. In some cases, e.g., Clarity [40] and WIG [2], where the calculation of the probability of a token in the table or corpus is required, neural aggregated matching feature computes the cosine similarity between the token vector and all the table or corpus token vectors first, and then aggregates those through a maximum function.

3.2.3. Neural Distance (ND) Feature

An alternative approach to estimate the relevance of a table and a query is based on the distance between them [10, 57] by considering the fact that the lower the distance between a query and a table is, the greater their similarity would be and the more relevant they would be to each other. A recent method, based on Earth Mover's Distance (EMD) [58], involves estimating the similarity between pairs of documents by minimizing the amount of distance that each embedded term in the first document needs to 'travel' to reach the embedded terms in the second document. This measure, referred to as the Word Mover's Distance (WMD), was proposed by Kusner et al. [59]. In this feature, given T and q, the distance between them will be based on tokens traveling from q to tokens in T. As defined in [59], the transportation matrix θ is a flow matrix in which $\theta_{i,j}$ shows to what degree token i in q is transported to token j in T. Matrix $\theta_{i,j}$, which essentially determines what token pairs from the two documents should be connected to each other, needs to be learnt

Table 3: The neural distance feature integrated into post-retrieval query performance predictors. Table format is similar to Table 1. Also, $ND(t_C, \pi^k_R(q;C) = \sum_{T \in \pi^k_R(q;C)} ND(t_C, T)(\prod_{t_q \in \varphi(q)} ND(t_q, T))$.

Base QPP Method	P^{ND}_{base}: Revised Post-retrieval Query Performance Predictors with Neural Distance Feature *(ND)*
Clarity [40]	$P^{ND}_{Clarity}(\varphi(\pi^k_R(q;C))) = \sum_{t_C \in V(\varphi(C))} ND(t_C, \pi^k_R(q;C)) \log \frac{ND(t_C, \pi^k_R(q;C))}{ND(t_C,C)}$
WIG [2]	$P^{ND}_{WIG}(\varphi(\pi^k_R(q;C))) = \frac{1}{k} \sum_{T \in \pi^k_R(q;C)} \sum_{t_q \in \varphi(q)} \lambda(t_q) \log \frac{ND(t_q,T)}{ND(t_q,C)}$
NQC [3]	$P^{ND}_{NQC}(\varphi(\pi^k_R(q;C))) = \frac{\sqrt{\frac{1}{k} \sum_{T \in \pi^k_R(q;C)} (ND(q,T) - \mu)^2}}{\lvert ND(q,C) \rvert}$
SMV [44]	$P^{ND}_{SMV}(\varphi(\pi^k_R(q;C))) = \frac{\sqrt{\frac{1}{k} \sum_{T \in \pi^k_R(q;C)} (ND(q,T) \lvert \ln \frac{ND(q,T)}{\mu} \rvert)}}{ND(q,C)}$

based on a linear optimization program. The distance between two documents can be calculated by minimizing the following linear optimization function:

$$WMD(q,T) = \sum_{i=1}^{|\varphi(q)|} \sum_{j=1}^{|\varphi(T)|} \theta_{i,j} d(i,j) \tag{5}$$

where $d(i,j)$ is the distance between the vector of token i in q and the vector of token j in T in the embedding space. Our proposed Neural Distance (ND) feature estimates the relevance degree of table T to query q based on the minimum value computed from optimizing Equation 5. As such, we formally define the Neural Distance (ND) feature as follows:

Definition 3. ***(Neural Distance Feature)*** *Given q and T, neural distance feature is equivalent to the Word Mover's Distance between the vectors of tokens in q and the vectors of tokens in T, denoted as $ND(q,T)$.*

Table 3 shows how to integrate the Neural Distance feature in post-retrieval predictors.

4. Experiments

The objective of our experiments is to evaluate the effectiveness of our proposed neural predictors on the task of query performance prediction for ad hoc table retrieval. To this end, we define four research questions that will be answered through performing empirical experimentation. More specifically, we address the following research questions in our experiments:

RQ1 Given the *Representation function* φ, which can provide tokens in the form of either keywords or entities to represent queries and tables, which type of token would be most effective for determining query difficulty?

RQ2 Based on the three proposed neural features, would it be possible to improve the estimation power of existing post-retrieval query performance predictors?

RQ3 Would the proposed neural predictors have a complementary impact on existing QPP methods and hence lead to improved performance when systematically interpolated with baseline methods?

RQ4 Whether the consideration of table specific structural characteristics such as table cells, rows and columns have any impact on the performance of our proposed neural predictors?

Summarily, RQ1 will be investigating which form of query, table and corpus representation, through either keywords or entities, be more effective in the context of the QPP task. RQ2 explores whether the integration of the proposed neural features into existing QPP methods leads to improved performance prediction. RQ3 will explore the synergistic nature of the neural features and existing keyword-based QPP methods to see whether their interpolation will lead to improved performance and RQ4 investigates whether the explicit consideration of table characteristics will lead to improved performance by the QPP method.

In the following, we first introduce the experimental settings, including the ranking functions, the test collection and the evaluation metrics. We then report and analyze our results and present the most important findings in order to answer the research questions.

4.1. Test Collection

We employed the test collection introduced in [14] in our experiments. This test collection includes three main components, namely the table corpus, the query set and the relevance judgments. The table corpus is composed of the WikiTables corpus [57] containing over 1.6M tables. For each table, five information fields are provided: table caption, column headings, data rows, page title,

and section title. To represent the tables as the bag of entities, we use the same entities that were extracted by Zhang and Balog [14]. The query set consists of 60 queries, which were sampled from two independent sources. We also used query entities as provided in [15]. The relevance judgments consist of 3,120 query-table pairs. Out of these, 377 are labeled as highly relevant, 474 as relevant, and 2,269 as non-relevant.

4.2. Ranking Functions

Given a query q, a ranking function R assigns a real number to each table $T \in C$ that represents the relative relevance of T with respect to q. In the experiments, we use seven different ranking functions to serve as R. These ranking functions can be broadly classified as (1) *content-based ranking functions* that focus on table content for measuring relevance between the query and table spaces; and (2) *feature-based ranking functions* that define table and query characteristics that take the structural characteristics of tables into account.

1. Content-based Ranking Functions:

   (a) STR ranking [14] models both the table and the query as sets of semantic vectors and then uses two general strategies (early and late fusion), for computing the similarity between queries and tables based on their semantic representations.

   (b) Single Field ranking [60] represents tables as an ordinary document and adopts ad hoc document retrieval, in the form of a language model with Dirichlet smoothing, for ranking tables.

   (c) Table2VecW and Table2VecE [17] are content-based ranking functions that learn neural representations for tables. The difference between the two ranking functions is based on the content they consider in the table to learn the neural representation. Table2VecW benefits from title, section title, table caption, table headings, and all table cells, while Table2VecE employs the sequence of all entities that appear within table cells.

2. Feature-based Ranking Functions:

   (a) LTR ranking [14] uses the full set of features listed in [23, 24] and trains a Random Forest [61] regression as the learner with 1,000 trees.

Table 4: The state-of-the-art post-retrieval query performance predictors used as baselines in our work. T is a table in corpus C and $V(\varphi(C))$ is the vocabulary of the unique tokens in the corpus C. $\lambda(t_q)$ reflects the relative weight of the token's type t_q and is inversely related to the square root of the number of query tokens of that type. μ is the mean score of tables in $\pi^k_R(q;C)$. $\Pr\left(t_C \,\middle|\, \pi^k_R(q;C)\right) = \sum_{T \in \pi^k_R(q;C)} \Pr\left(t_C \,|\, T\right) \left(\prod_{t_q \in \varphi(q)} \Pr\left(t_q \,|\, T\right)\right)$.

Baseline Predictors	Formula
Clarity [40]	$Clarity(\varphi(\pi^k_R(q;C))) = \sum_{t_C \in V(\varphi(C))} \Pr\left(t_C \,\middle|\, \pi^k_R(q;C)\right) \log \frac{\Pr\left(t_C \,\middle|\, \pi^k_R(q;C)\right)}{\Pr(t_C \,|\, C)}$
WIG [2]	$WIG(\varphi(\pi^k_R(q;C))) = \frac{1}{k} \sum_{T \in \pi^k_R(q;C)} \sum_{t_q \in \varphi(q)} \lambda(t_q) \log \frac{\Pr(t_q \,|\, T)}{\Pr(t_q \,|\, C)}$
NQC [3]	$NQC(\varphi(\pi^k_R(q;C))) = \frac{\sqrt{\frac{1}{k} \sum_{T \in \pi^k_R(q;C)} (Score(q,T) - \mu)^2}}{\lvert Score(q,C) \rvert}$
SMV [44]	$SMV(\varphi(\pi^k_R(q;C))) = \frac{\sqrt{\frac{1}{k} \sum_{T \in \pi^k_R(q;C)} (Score(q,T) \lvert \ln \frac{Score(q,T)}{\mu} \rvert)}}{Score(q,C)}$

(b) WebTable ranking [24] employs the features mentioned in [24] for each table and trains linear regression as the ranking function with 5-fold cross validation.

(c) WikiTable ranking [23] considers a Lasso model [62] with coordinate ascent as the ranking function, which is trained on the features from [23] with 5-fold cross validation.

We note that the selected ranking functions are either (1) state of the art (SOTA) methods in ad hoc table retrieval such as STR and Table2VecW and Table2VecE, (2) have been used as baseline in several recent publications such as WikiTable and WebTable or (3) are standard information retrieval methods that have shown robust retrieval performance across different tasks such as Single Field. The run files for all the ranking functions, including content-based and feature-based functions are readily available[1]. Accordingly, the result list length per query provided by the run

[1]The runs for STR, LTR, WikiTable, WebTable and Single Field ranking functions can be obtained from

Table 5: Performance of the ranking function on the ad hoc table retrieval task in terms of NDCG@10 and @20.

	Method	NDCG@10	NDCG@20
Content-based	STR	0.6293	0.6825
	Single Field	0.4860	0.5473
	Table2VecW	0.6096	0.6505
	Table2VecE	0.5569	0.6161
Feature-based	LTR	0.5456	0.6031
	WikiTable	0.4766	0.5206
	WebTable	0.2992	0.3726

files is 20. We report the performance of the ranking functions based on the run files in Table 5.

4.3. Evaluation Metrics

The prediction quality of a QPP method is evaluated by measuring Pearson and Kendall Correlations between the predicted ranking of the queries against the actual ranking of the queries according to their observed performance measured on $NDCG$ and AP [63].

4.4. Baselines

For comparative analytics, we adopt Clarity [40], WIG [2], NQC [3] and SMV [44], which are among the most widely used query performance predictors. The detailed formulation of these predictors is provided in Table 4. We note that these metrics are implemented according to the detailed description provided in [37] and do not include any relation to neural embeddings.

Unlike these four baseline predictors that are based on a completely unsupervised formulation of the predictor, the most recent post-retrieval query performance predictor is work by Zamani et al [48] that uses a weak supervision strategy to learn query rankings. As such, while we will compare our work with this baseline QPP method, we cannot use it to integrate our approach with it.

It should be noted that given our approach requires neural embeddings, in our experiments, we use the pre-trained embedding vectors from [64] for neural embeddings of entities and the Google

https://github.com/iai-group/www2018-table while the runs for Table2VecW and Table2VecE can be accessed at https://github.com/iai-group/sigir2019-table2vec.

News embeddings[2] for neural embeddings of keywords. For BERT, we use the pretrained model offered by HuggingFace with 12 layers, 768 hidden layers, 12 heads, and 110M parameters trained on cased English text, often known as BERT-Base Cased. For ELMo, we have used the model provided by Flair [65].

4.5. Analysis

We present the findings of our experiments based on the four research questions introduced earlier.

4.5.1. RQ1. Impact of the Representation Function φ

The objective of the first research question is to determine which token representation is most suitable for modeling tables and queries when performing query performance prediction in the context of ad hoc table retrieval. The representation function φ can represent tables and queries in the form of keywords or entities. The idea behind a bag of words representation is that lexical similarity between tables and queries can be considered to be an indication of relevance. On the other hand, the intuition behind the bag of entities representation is that the relevance of a table to a query can be captured beyond mere keywords and can depend on the semantic association between the two, which might not necessarily have much lexical overlap. Bagheri and Al-Obeidat have already shown this by providing examples of where queries and tables that do not have any lexical overlap are considered to be relevant as assessed in human judgements [15]. For instance, for Query 30, 'pain medication', in the query set of our test collection, the human judgements had determined that Table 0520-188 'diseases and condition' was relevant but the query and table did not have any shared keywords. Therefore, there is existing evidence to support that the representation of tables as bag of entities can be beneficial. In this research question, we are interested in understanding the impact of each representation on query performance prediction.

To this end, we compare the performance of our neural predictors on different ranking functions by using two different representations: bag of words and bag of entities. The results are shown in Tables 6-9. The tables present the results for NDCG and AP for both Kendall and Pearson correlations. In these tables, the best predictor for the combination of each QPP method and

[2]https://code.google.com/archive/p/word2vec/

ranking function is shown in bold. As seen in the table, performances for both measures, i.e., NDCG and AP, as well as correlations, i.e., Kendall and Pearson, are consistent.

Table 6: The results of various neural predictors with different ranking functions for both bag of words and bag of entities based representations in terms of Pearson Correlation. The bold values show the largest value in each rectangle related to a ranking function and a QPP method. For instance, **0.172** is largest value related to STR and Clarity.

$NDCG$@20 Pearson		Ranking Function													
		Content-based								Feature-based					
		STR		Single Field		Table2VecW		Table2VecE		WikiTable		WebTable		LTR	
		Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity
Clarity	$P^{NM}_{Clarity}$	0.138	0.114	**0.237**	0.106	0.154	**0.204**	0.129	0.168	0.169	**0.228**	0.159	0.244	0.034	**0.167**
	$P^{NAM}_{Clarity}$	**0.172**	0.115	0.2	0.097	0.168	0.188	0.032	**0.18**	0.025	0.186	0.039	**0.25**	0.098	0.142
	$P^{ND}_{Clarity}$	0.051	0.068	0.115	0.094	0.059	0.099	0.059	0.05	0.031	0.026	0.079	0.036	0.018	0.069
WIG	P^{NM}_{WIG}	0.085	0.087	0.009	0.106	0.009	0.126	0.043	0.119	0.025	0.136	0.049	0.131	0.003	0.053
	P^{NAM}_{WIG}	**0.387**	0.021	**0.457**	0.101	**0.339**	0.046	**0.389**	0.031	**0.42**	0.034	0.111	**0.177**	**0.341**	0.008
	P^{ND}_{WIG}	0.032	0.016	0.103	0.056	0.009	0.169	0.013	0.119	0.182	0.042	0.037	0.032	0.06	0.087
NQC	P^{NM}_{NQC}	**0.177**	0.007	0.107	0.172	0.032	0.071	0.017	0.031	0.063	0.059	0.003	0.097	0.114	0.062
	P^{NAM}_{NQC}	0.015	0.025	**0.173**	0.16	0.173	0.056	**0.109**	0.012	**0.285**	0.059	0.089	**0.116**	**0.163**	0.06
	P^{ND}_{NQC}	0.08	0.021	0.08	0.112	**0.228**	0.173	0.01	0.01	0.006	0.081	0.1	0.075	0.021	0.062
SMV	P^{NM}_{SMV}	**0.161**	0.006	0.115	0.203	0.006	0.055	0.013	0.01	0.025	0.091	0.009	0.088	0.087	0.039
	P^{NAM}_{SMV}	0.042	0.029	0.136	**0.244**	0.175	0.085	**0.063**	0.012	**0.266**	0.11	**0.131**	0.06	**0.164**	0.011
	P^{ND}_{SMV}	0.053	0.02	0.085	0.063	**0.306**	0.161	0.017	0.007	0.029	0.124	0.079	0.112	0.003	0.02

Table 7: The results of various neural predictors with different ranking functions for both bag of words and bag of entities based representations in terms of Pearson Correlation. The bold values show the largest value in each rectangle related to a ranking function and a QPP method.

AP@20 Pearson		Ranking Function													
		Content-based								Feature-based					
		STR		Single Field		Table2VecW		Table2VecE		WikiTable		WebTable		LTR	
		Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity
Clarity	$P^{NM}_{Clarity}$	0.156	0.114	**0.242**	0.115	0.182	**0.257**	0.17	**0.196**	0.157	**0.22**	0.199	**0.278**	0.013	**0.207**
	$P^{NAM}_{Clarity}$	**0.186**	0.104	0.189	0.094	0.172	0.188	0.067	0.163	0.01	0.196	0.066	0.272	0.049	0.157
	$P^{ND}_{Clarity}$	0.131	0.102	0.152	0.218	0.136	0.031	0.167	0.022	0.036	0.079	0.14	0.078	0.004	0.095
WIG	P^{NM}_{WIG}	0.03	0.096	0.066	0.114	0.042	0.153	0.002	0.168	0.008	0.136	0.064	0.162	0.021	0.068
	P^{NAM}_{WIG}	**0.362**	0.012	**0.4**	0.055	**0.32**	0.015	**0.365**	0.036	**0.388**	0.047	0.097	0.23	**0.327**	0.017
	P^{ND}_{WIG}	0.065	0.049	0.151	0.01	0.018	0.187	0.024	0.152	0.137	0.039	0.073	**0.245**	0.037	0.116
NQC	P^{NM}_{NQC}	**0.19**	0.003	0.134	0.152	0.005	0.037	0.007	0.04	0.022	0.013	0.056	0.092	0.047	0.035
	P^{NAM}_{NQC}	0.113	0.028	**0.162**	0.128	**0.306**	0.042	**0.192**	0.009	**0.317**	0.033	**0.17**	0.095	**0.277**	0.022
	P^{ND}_{NQC}	0.103	0.013	0.087	0.085	0.209	0.15	0.007	0.006	0.051	0.039	0.152	0.084	0.009	0.025
SMV	P^{NM}_{SMV}	**0.171**	0.0001	0.143	0.185	0.021	0.019	0.018	0.021	0.07	0.051	0.058	0.089	0.015	0.01
	P^{NAM}_{SMV}	0.074	0.039	0.1	**0.218**	**0.288**	0.067	**0.12**	0.014	**0.301**	0.081	**0.221**	0.054	**0.26**	0.021
	P^{ND}_{SMV}	0.108	0.005	0.11	0.12	0.274	0.143	0.006	0.019	0.084	0.083	0.134	0.126	0.018	0.008

Table 8: The results of various neural predictors with different ranking functions for both bag of words and bag of entities based representations in terms of Kendall Correlation. The bold values show the largest value in each rectangle related to a ranking function and a QPP method.

$NDCG$@20 Kendall		Ranking Function													
		Content-based								Feature-based					
		STR		Single Field		Table2VecW		Table2VecE		WikiTable		WebTable		LTR	
		Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity
Clarity	$P^{NM}_{Clarity}$	0.094	0.127	0.153	0.018	**0.188**	0.134	**0.177**	0.165	**0.134**	0.111	0.078	0.187	0.084	0.123
	$P^{NAM}_{Clarity}$	0.087	0.117	0.156	0.047	0.139	0.058	0.06	0.064	0.005	0.099	0.127	**0.21**	0.067	0.097
	$P^{ND}_{Clarity}$	**0.144**	0.045	**0.208**	0.017	0.144	0.012	0.133	0.001	0.107	0.003	0.078	0.053	**0.158**	0.106
WIG	P^{NM}_{WIG}	0.109	0.043	0.009	0.006	0.036	0.115	0.016	0.037	0.019	0.0001	0.003	0.119	0.046	0.056
	P^{NAM}_{WIG}	**0.234**	0.048	**0.28**	0.082	**0.26**	0.081	**0.264**	0.043	**0.2**	0.072	0.019	0.122	**0.212**	0.042
	P^{ND}_{WIG}	0.055	0.0001	0.1	0.004	0.005	0.074	0.024	0.045	0.146	0.064	0.033	**0.142**	0.03	0.004
NQC	P^{NM}_{NQC}	**0.081**	**0.081**	0.036	**0.207**	0.034	0.09	0.008	0.009	0.046	0.034	0.041	0.068	0.077	0.036
	P^{NAM}_{NQC}	0.06	0.069	0.137	0.2	**0.13**	0.067	**0.148**	0.063	**0.153**	0.075	0.088	0.081	**0.16**	0.024
	P^{ND}_{NQC}	0.014	0.036	0.004	0.134	0.061	0.104	0.031	0.008	0.037	0.038	0.095	**0.1**	0.031	0.067
SMV	P^{NM}_{SMV}	0.042	0.063	0.029	0.214	0.006	0.077	0.01	0.008	0.028	0.072	0.054	0.067	0.068	0.033
	P^{NAM}_{SMV}	0.049	**0.071**	0.13	**0.242**	**0.136**	0.108	**0.114**	0.037	**0.167**	0.088	0.112	0.032	**0.161**	0.014
	P^{ND}_{SMV}	0.024	0.041	0.006	0.181	0.1	0.108	0.041	0.006	0.032	0.08	0.079	**0.125**	0.027	0.026

Table 9: The results of various neural predictors with different ranking functions for both bag of words and bag of entities based representations in terms of Kendall Correlation. The bold values show the largest value in each rectangle related to a ranking function and a QPP method.

AP@20 Kendall		Ranking Function													
		Content-based								Feature-based					
		STR		Single Field		Table2VecW		Table2VecE		WikiTable		WebTable		LTR	
		Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity
Clarity	$P^{NM}_{Clarity}$	0.087	**0.144**	0.166	0.049	**0.206**	0.162	**0.195**	0.174	0.104	0.17	0.101	0.17	0.087	0.135
	$P^{NAM}_{Clarity}$	0.079	0.048	0.151	0.023	0.126	0.087	0.053	0.025	0.007	**0.184**	0.115	**0.184**	0.078	0.056
	$P^{ND}_{Clarity}$	0.128	0.086	**0.202**	0.001	0.146	0.068	0.166	0.058	0.173	0.038	0.117	0.038	**0.187**	0.099
WIG	P^{NM}_{WIG}	0.016	0.056	0.041	0.0001	0.028	0.159	0.001	0.086	0.026	0.119	0.007	0.119	0.018	0.06
	P^{NAM}_{WIG}	**0.225**	0.004	**0.232**	0.059	**0.252**	0.063	**0.251**	0.005	**0.258**	0.133	0.033	0.133	**0.235**	0.028
	P^{ND}_{WIG}	0.048	0.03	0.117	0.007	0.019	0.099	0.022	0.058	0.091	0.176	0.065	**0.176**	0.019	0.037
NQC	P^{NM}_{NQC}	**0.143**	0.032	0.093	**0.166**	0.013	0.034	0.002	0.003	0.02	0.082	0.038	0.082	0.049	0.02
	P^{NAM}_{NQC}	0.127	0.041	0.095	0.143	**0.224**	0.016	**0.145**	0.05	**0.193**	0.083	**0.124**	0.083	**0.205**	0.007
	P^{ND}_{NQC}	0.063	0.002	0.028	0.108	0.083	0.082	0.038	0.04	0.054	0.079	0.094	0.079	0.018	0.034
SMV	P^{NM}_{SMV}	0.112	0.027	0.094	0.175	0.032	0.026	0.011	0.011	0.051	0.081	0.052	0.081	0.034	0.018
	P^{NAM}_{SMV}	**0.119**	0.06	0.074	**0.197**	**0.225**	0.057	**0.116**	0.047	**0.209**	0.042	**0.149**	0.042	**0.199**	0.034
	P^{ND}_{SMV}	0.083	0.03	0.05	0.124	0.103	0.098	0.04	0.017	0.058	0.102	0.078	0.102	0.029	0.002

First, we compare the quality of our proposed neural predictors for both bag of words and bag of entities representations. The results reported in Tables 6-9 show that while the bag of entities representation is computationally expensive, it results in weaker quality compared to the bag of words representation for most neural predictors and ranking functions, regardless of being content-based or feature-based (except for the WebTable ranking function). The main reason that we find for this is that it may be due to the fact that many of the tables in the test collection do not contain a noticeable number of entities. This was a finding by Bagheri and Al-Obeidat [15] as well who reported that an entity-based retrieval method is only effective for those tables that consist of a sizeable number of entities; otherwise, a sparse entity representation for the tables, resulting from the lack of a sufficient number of entities, would lead to poor performance. Similarly to [15], we find that the lack of entities to represent tables leads to poorer performance by our proposed neural predictors when using the bag of entities representation.

We further study the effectiveness of bag of words and bag of entities based representations on two state-of-the-art post-retrieval QPP methods in Table 10, namely Clarity and WIG. We cannot include SMV and NQC in this table as they rely on table relevance scores and do not depend on whether the table representation is based on entities or keywords. As such, they are not applicable to RQ1. As seen in this table, in the majority of the cases, the bag of words based representation shows a superior performance compared to the bag of entities based representation on both NDCG and AP for both Kendall and Pearson correlations. We again note consistent observations across NDCG and AP for both correlation measures. Therefore, we conclude that in the context of ad hoc table retrieval, and for both when our neural predictors are present and otherwise, using a bag of words representation for both tables and queries is sufficient for performing the QPP task.

Finding 1. In the context of ad hoc table retrieval, a bag of words representation of tables and queries is the more efficient form of representation compared to the bag of entities representation as it provides stronger performance for the QPP task.

Table 10: The results of baseline predictors with different ranking functions for both bag of words and bag of entities based representations in terms of Pearson and Kendall Correlation on both NDCG and AP. Bold values show the larger value based on the pair of ranking fuction and QPP method for word vs entity.

			Ranking Function													
			Content-based								Feature-based					
			STR		Single Field		Table2VecW		Table2VecE		WikiTable		WebTable		LTR	
			Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity	Word	Entity
Pearson	*NDCG*@20	*Clarity*	**0.174**	0.006	**0.14**	0.063	0.051	**0.072**	**0.285**	0.037	0.004	**0.108**	**0.17**	0.003	**0.293**	0.157
		WIG	**0.209**	0.004	**0.275**	0.0001	**0.149**	0.075	**0.283**	0.08	**0.205**	0.053	0.2	**0.224**	**0.272**	0.119
	AP@20	*Clarity*	**0.202**	0.027	**0.157**	0.06	0.029	**0.098**	**0.244**	0.06	0.082	**0.141**	**0.191**	0.022	**0.263**	0.175
		WIG	**0.231**	0.032	**0.353**	0.071	**0.197**	0.143	**0.283**	0.089	**0.232**	0.028	0.194	**0.211**	**0.291**	0.137
Kendall	*NDCG*@20	Clarity	**0.079**	0.055	**0.041**	0.01	0.055	**0.094**	**0.119**	0.045	0.003	**0.074**	**0.094**	0.05	**0.166**	0.031
		WIG	**0.098**	0.056	**0.176**	0.057	**0.089**	0.015	**0.171**	0.024	**0.133**	0.028	**0.106**	0.067	**0.16**	0.04
	AP@20	*Clarity*	**0.08**	0.015	**0.02**	0.011	0.062	**0.063**	**0.131**	0.033	0.011	**0.047**	**0.106**	0.069	**0.153**	0.013
		WIG	**0.148**	0.047	**0.239**	0.002	**0.158**	0.043	**0.176**	0.011	**0.169**	0.02	**0.107**	0.051	**0.196**	0.027

4.5.2. RQ2. Performance of the Neural Predictors

In the second research question, we aim at exploring the impact of the proposed neural features on query performance prediction. We expect that the impact of the neural predictors would be different for the various ranking functions especially because the ranking functions consider a different set of features when determining relevance and performing ranking, i.e., content-based ranking functions compared to feature-based ranking functions. For instance, a content-based ranking function such as STR [14] measures the similarity between tables and queries based on their semantic associations, which means that such a method would be able to retrieve relevant yet lexically non-overlapping tables for an input query. In contrast, a feature-based ranking method such as LTR [14] is based on a set of structural features and hence would be sensitive to how information is spread within the table structure. As such, given the fact that the performance of the ranking functions are different, we expect that the performance of the QPP methods would also be different.

In our experiments, we compare the performance of the four baseline QPP methods and our proposed neural predictors over all ranking functions. The results are reported in Table 11. In this table, the best performing QPP method for each of the ranking functions has been shown in Bold. Furthermore, given the findings from RQ1, we report the results of the QPP methods based on the bag of words representation noting the weaker performance of the bag of entities representation for this task. We also note that we resort to reporting NDCG using Kendall correlation as we already showed in Tables 6-9 that AP and NDCG values with both Kendall and Pearson correlations have consistent behavior.

Our first observation is that our proposed P_{WIG}^{NAM} predictor has the best performance for content-based ranking functions, i.e., the STR, Single Field, Table2VecW and Table2VecE ranking functions. In contrast, in the feature-based ranking functions, the baseline predictors without the integration of our proposed neural features are more successful. This observation confirms the relationship between the query performance predictors and the characteristics of the ranking functions for bag of word based representation in the table retrieval task. The impact of our proposed neural predictors are most visibly observed when the ranking functions consider the similarity between the *contents* of tables and the queries. Given our proposed neural predictors employ neural embeddings to capture the semantic association between keyword vectors and the distances be-

Table 11: The quality of different query performance predictors for each ranking function in terms of Kendall's correlation. The best value for each ranking function is specified in boldface.

$NDCG$@20 Kendall		Ranking Function						
		Content-based				Feature-based		
		STR	Single Field	Table2VecW	Table2VecE	WikiTable	WebTable	LTR
Clarity	$Clarity$	0.079	0.041	0.055	0.119	0.003	0.094	0.166
	$P^{NM}_{Clarity}$	0.094	0.153	0.188	0.177	0.134	0.078	0.084
	$P^{NAM}_{Clarity}$	0.087	0.156	0.139	0.06	0.005	0.127	0.067
	$P^{ND}_{Clarity}$	0.144	0.208	0.144	0.133	0.107	0.078	0.158
WIG	WIG	0.098	0.176	0.089	0.171	0.133	0.106	0.16
	P^{NM}_{WIG}	0.109	0.009	0.036	0.016	0.019	0.003	0.046
	P^{NAM}_{WIG}	**0.234**	**0.28**	**0.26**	**0.264**	0.2	0.019	0.212
	P^{ND}_{WIG}	0.055	0.1	0.005	0.024	0.146	0.033	0.03
NQC	NQC	0.076	0.041	0.229	0.252	**0.318**	0.165	**0.338**
	P^{NM}_{NQC}	0.081	0.036	0.034	0.008	0.046	0.041	0.077
	P^{NAM}_{NQC}	0.06	0.137	0.13	0.148	0.153	0.088	0.16
	P^{ND}_{NQC}	0.014	0.004	0.061	0.031	0.037	0.095	0.031
SMV	SMV	0.084	0.055	0.219	0.239	0.303	**0.17**	0.308
	P^{NM}_{SMV}	0.042	0.029	0.006	0.01	0.028	0.054	0.068
	P^{NAM}_{SMV}	0.049	0.13	0.136	0.114	0.167	0.112	0.161
	P^{ND}_{SMV}	0.024	0.006	0.1	0.041	0.032	0.079	0.027

tween the embedded keyword vectors, they are able to model and capture semantic relationships, which are primarily content-based in nature. This reflects itself in developing stronger QPP methods for content-based ranking functions that are based on query and table semantic associations or similarity through language models.

Finding 2. For the purpose of predicting the performance of ranking functions that are primarily based on the *content value* of the tables (content-based ranking functions), while the neural predictors do not always improve the performance of the baselines, the integration of one of our proposed neural predictors with the WIG metric leads to consistent and noticeable performance improvement over all baselines.

For the feature-based ranking functions such as LTR, WikiTable and WebTable that are primarily based on structural features of the table, the base QPP methods without the integration of our proposed neural features show better performance. The structural features of the table include features, such as the number of rows, columns, and empty cells, and the query word frequency in specific parts of the table, such as the leftmost column, or second-to-left column, that cannot be effectively captured by our proposed neural predictors. As such, the baseline QPP methods show better performance compared to our proposed neural predictors.

Finding 3. Traditional QPP methods that do not account for the semantic association between the table and query spaces are more effective for determining query difficulty of feature-based ranking functions that consider table-specific structural features.

As mentioned earlier, one of the strongest state of the art post-retrieval query performance prediction methods in the literature for document retrieval is the work by Zamani et al [48], which uses weak supervision to rank queries. Given this method requires training and is not fully unsupervised, our proposed predictors cannot be integrated into its formulation. However, we compare our work against this baseline as well in Table 12. For easier comparison to the performance of our own work, we have included P^{NAM}_{WIG} in this table; however, comparison against other variations of our work can be done according to performance values reported in Tables 6-9. As shown in Table 12, the only case where the weakly supervised method by Zamani et al shows better performance compared to P^{NAM}_{WIG} is on the STR method (and only on Pearson correlation). However, for all other ranking functions, it shows a consistently weaker performance. We also

Table 12: Comparative performance of the work by Zamani et al [48] with P_{WIG}^{NAM}. Please note P stands for Pearson and K stands for Kendall correlation. The only places where the baseline is better than our proposed approach is marked with boldface.

		P_{WIG}^{NAM}				Zamani et al			
		NDCG@20		AP@20		NDCG@20		AP@20	
		P	K	P	K	P	K	P	K
Content-based	STR	0.387	0.234	0.362	0.225	0.322	**0.295**	0.338	**0.271**
	Single Field	0.457	0.280	0.400	0.232	0.289	0.209	0.307	0.228
	Table2VecW	0.339	0.260	0.320	0.252	0.104	0.089	0.011	0.001
	Table2VecE	0.389	0.264	0.365	0.251	0.037	0.009	0.087	0.041
	WikiTable	0.420	0.200	0.388	0.258	0.060	0.086	0.010	0.016
Feature-based	WebTable	0.111	0.019	0.097	0.033	0.124	0.085	0.113	0.073
	LTR	0.341	0.212	0.327	0.235	0.110	0.116	0.150	0.138

note that while all the baseline QPP methods as well as our proposed neural predictors are fully unsupervised, the work proposed in [48] requires a training process but even so, while shown to be effective for QPP in document retrieval, it is not as effective for the context of table retrieval.

4.5.3. RQ3. Synergistic Impact of the Neural Predictors

The main objective of the third research question is to investigate whether the interpolation of neural predictors with base QPP methods will enhance the quality of the state-of-the-art predictors or not. Our hypothesis is that the neural and baseline predictors can produce overlapping results while in many cases a subset of their correct results is distinct and non-overlapping. For this reason, we believe that the interpolation of these predictors can result in synergy between the correctly identified results of each predictor and hence lead to improved quality. For this purpose, we employ the CombSum approach [45] and form three variations, referred to as (1) *Best Baseline + Best Neural*, (2) *Best of each Predictor - Best Neural + Baseline* and (3) *Best Word + Best Entity*. In all of three variations, first, the scores generated by each predictor is normalized and then, linear interpolation is performed.

Best Baseline + Best Neural. In this variation of our interpolations, for each ranking function, the best neural predictor and the best baseline QPP method are selected to be interpolated

Table 13: The evaluation results for the interpolation of the best neural predictor with the best baseline for each ranking function individually.

$NDCG$@20 Kendall	Ranking Function	Best Baseline	Best Neural Predictors	CumbSum	$\Delta_{Baseline}$	Δ_{Neural}
Content-based	STR	WIG	P^{NAM}_{WIG}	0.169	7.1%	-6.5%
	Single Field	WIG	P^{NAM}_{WIG}	0.269	9.3%	-1.1%
	Table2VecW	NQC	P^{NAM}_{WIG}	0.257	2.8%	-0.3%
	Table2VecE	NQC	P^{NAM}_{WIG}	0.298	4.6%	3.4%
Feature-based	WikiTable	NQC	P^{NAM}_{WIG}	0.311	-0.7%	11.1%
	WebTable	SMV	$P^{NAM}_{Clarity}$	0.1273	-4.27%	0.03%
	LTR	NQC	P^{NAM}_{WIG}	0.311	-2.7%	9.9%

through CombSum. The quality of the final CombSum predictor in terms of Kendall correlation and the amount of achieved improvement over the baseline ($\Delta_{Baseline}$) and neural predictors (Δ_{Neural}) are reported in Table 13. The first observation based on $\Delta_{Baseline}$ is that the best neural predictors have been able to improve the best baselines that they have been interpolated with in content-based ranking functions. Another observation here is with regards to the performance of the interpolated QPP method for content-based ranking functions. The observed performance is consistent for these ranking functions where the best neural predictor is P^{NAM}_{WIG} for all content-based methods. Furthermore, the interpolation has been able to consistently improve the best baseline QPP method.

This indicates that:

1. the best non-neural baseline QPP methods cannot provide synergistic support for our proposed neural predictors when applied to methods that are solely based on table and query content (content-based ranking);

2. the proposed neural predictors in this paper are able to provide additional complementary performance over baseline QPP methods when dealing with feature-based ranking functions.

Finding 4. The linear interpolation of the best baseline QPP method with our best neural predictor leads to improved performance over the best baseline QPP method for content-based ranking functions. In contrast, improvements are observed over the best neural predictors when the ranking functions are feature-based in nature (LTR, WebTable, WikiTable).

An important point to mention here based on $\Delta_{Baseline}$ is that the degree of improvement over the baselines in content-based ranking function is more than other ranking functions and we conclude that the neural predictor is more effective in increasing the quality of the baselines when dealing with content-based ranking functions. This confirms our findings with regards to the relationship between the ranking functions and query performance predictors concluded in RQ2.

Best of each Predictor - Best Neural + Baseline. In this variation of our interpolations, we study the performance of interpolation for each predictor individually. For this purpose, in each ranking function, the best neural predictor is selected to be interpolated with the baseline of the same predictor through the CombSum approach. Table 14 shows the interpolation performance and the improvement percentages over the baseline and neural predictor for Clarity. As shown in Table 14, interpolating the best neural predictor with Clarity will not lead to improved performance over the Clarity method itself in WebTable. The results from Δ_{Neural} show that the quality of the best neural predictors are better than the interpolated methods (except on the STR and LTR ranking functions). Therefore, we find that the interpolation of our proposed neural predictors with Clarity can potentially lead to improvements over Clarity (except WebTable) but not over the neural predictors; therefore, showing non-synergistic overlapping behavior between them.

We report the quality of CombSum approach and the improvements over the best neural predictors and baseline for WIG in Table 15. The main observation here is that the neural aggregated matching feature when applied to WIG is the best neural predictor for all ranking functions (except WebTable). Another observation is that the interpolation of P^{NAM}_{WIG} with WIG improves the performance of WIG in all ranking functions except for WebTable. Also, the results show that when P^{NAM}_{WIG} is interpolated with WIG, the estimation quality of P^{NAM}_{WIG} is increased in feature-based ranking functions while this increase cannot be observed for content-based ranking functions.

NQC and SMV are score-oriented models and consider the variance of the returned table scores for measuring the performance of a query as described earlier in Table 4. Tables 16 and 17 show

Table 14: Evaluation results for Clarity.

$NDCG$@20 Kendall	Ranking Function	Best Neural Predictors	CumbSum	$\Delta_{Baseline}$	Δ_{Neural}
Content-based	STR	$P^{ND}_{Clarity}$	0.245	16.6%	10.1%
	Single Field	$P^{ND}_{Clarity}$	0.197	15.6%	-1.1%
	Table2VecW	$P^{NM}_{Clarity}$	0.147	9.2%	-4.1%
	Table2VecE	$P^{NM}_{Clarity}$	0.136	1.7%	-4.1%
Feature-based	WikiTable	$P^{NM}_{Clarity}$	0.057	5.4%	-7.7%
	WebTable	$P^{NAM}_{Clarity}$	0.058	-3.6%	-6.9%
	LTR	$P^{ND}_{Clarity}$	0.261	9.5%	10.3%

Table 15: Evaluation results for WIG.

$NDCG$@20 Kendall	Ranking Function	Best Neural Predictors	CumbSum	$\Delta_{Baseline}$	Δ_{Neural}
Content-based	STR	P^{NAM}_{WIG}	0.169	7.1%	-6.5%
	Single Field	P^{NAM}_{WIG}	0.269	9.3%	-1.1%
	Table2VecW	P^{NAM}_{WIG}	0.198	10.9%	-6.2%
	Table2VecE	P^{NAM}_{WIG}	0.248	7.7%	-1.6%
Feature-based	WikiTable	P^{NAM}_{WIG}	0.209	7.6%	0.9%
	WebTable	P^{ND}_{WIG}	0.049	-5.7%	1.6%
	LTR	P^{NAM}_{WIG}	0.229	6.9%	1.7%

Table 16: Evaluation results for NQC.

$NDCG$@20 Kendall	Ranking Function	Best Neural Predictors	CumbSum	$\Delta_{Baseline}$	Δ_{Neural}
Content-based	STR	P^{NM}_{NQC}	0.08	0.4%	-0.1%
	Single Field	P^{NAM}_{NQC}	0.085	4.4%	-5.2%
	Table2VecW	P^{NAM}_{NQC}	0.017	-21.2%	-11.3%
	Table2VecE	P^{NAM}_{NQC}	0.015	-23.7%	-13.3%
Feature-based	WikiTable	P^{NAM}_{NQC}	0.11	-20.8%	-4.3%
	WebTable	P^{ND}_{NQC}	0.107	-5.8%	1.2%
	LTR	P^{NAM}_{NQC}	0.048	-29%	-11.2%

Table 17: Evaluation results for SMV.

$NDCG$@20 Kendall	Ranking Function	Best Neural Predictors	CumbSum	$\Delta_{Baseline}$	Δ_{Neural}
Content-based	STR	P^{NAM}_{SMV}	0.041	-4.3%	-0.8%
	Single Field	P^{NAM}_{SMV}	0.116	6.1%	-1.4%
	Table2VecW	P^{NAM}_{SMV}	0.02	-19.9%	-11.6%
	Table2VecE	P^{NAM}_{SMV}	0.009	-23%	-10.5%
Feature-based	WikiTable	P^{NAM}_{SMV}	0.086	-21.7%	-8.1%
	WebTable	P^{NAM}_{SMV}	0.186	1.6%	7.4%
	LTR	P^{NAM}_{SMV}	0.014	-29.4%	-14.7%

Table 18: Evaluation results for the interpolation of the best predictors in the both word and entity representations with each other.

$NDCG$@20 Kendall	Ranking Function	Best Predictor in Word based Representation	Best Predictor in Entity based Representation	CumbSum	Δ_{Word}	Δ_{Entity}
Content-based	STR	P^{NAM}_{WIG}	$P^{NM}_{Clarity}$	0.23	-0.4%	10.3%
	Single Field	P^{NAM}_{WIG}	P^{NAM}_{SMV}	0.162	-11.8%	-4.5%
	Table2VecW	P^{NAM}_{WIG}	NQC	0.257	-0.3%	2.8%
	Table2VecE	P^{NAM}_{WIG}	NQC	0.298	3.4%	4.6%
Feature-based	WikiTable	NQC	NQC	0.318	-	-
	WebTable	SMV	$P^{NAM}_{Clarity}$	0.112	-5.8%	-9.8%
	LTR	NQC	NQC	0.338	-	-

the quality of CombSum after interpolation with the best neural predictor with NQC and SMV, respectively. For both NQC and SMV, the overall performance of CombSum is not comparable to our proposed neural baselines nor to other CombSums such as with WIG. As such, we conclude that the interpolation of the best neural predictors with NQC and SMV is not an effective strategy.

Finding 5. The linear interpolation of each base QPP method with its best neural predictor does not necessarily lead to improved performance over the base QPP method. However, the WIG method shows to be the strongest baseline method over all ranking functions, which is significantly improved as a result of interpolation with its best neural predictor (except WebTable).

Best Word + Best Entity. In the last variation of our interpolations, we investigate, while we already found that bag of words based representation is superior to bag of entities representation, whether the interpolation of the two representations leads to increased performance for QPP. To this end, the results of the best neural predictor for the bag of words and bag of entities representations are selected and interpolated through CombSum for each ranking function separately. Table 18 reports the degree of improvement over the best bag of words and bag of entities representations.

We observe that in most ranking functions, there are no consistent improvements observed over

the bag of words based representation or bag of entities based representation. We find there were no performance changes in LTR and WikiTable, which is due to the fact that the best predictors in the bag of words and bag of entities based representations are the same and therefore no improvements could be observed. Overall, when considering improvements obtained through the interpolation of the best predictors from bag of words and bag of entities representations in comparison with the best predictor from the bag of words representation, described by Δ_{Word} in Table 18, we conclude that the bag of entities representation does not contribute to meaningful improvements for the QPP task. This reinforces our findings in RQ1.

Finding 6. The best predictors for each of the ranking functions obtained through bag of words and bag of entities representations exhibit overlapping behavior and hence their interpolation does not lead to meaningful improvements over the best predictor from the bag of words representation.

We would like to report further on our findings by reviewing our observations across the three types of interpolations reported in this section. We find that the neural aggregated matching (NAM) feature achieves noticeable performance improvements when interpolated with baseline QPP methods. Based on the results of the first variation of our interpolations shown in Table 13, we can observe that the best neural predictor is built using NAM in seven of the ranking functions. This observation is confirmed based on the second variation of our interpolation that in most cases, NAM has been selected as the best neural predictor for the different base QPP methods. Finally, the results of the last variation show that NAM is the variation of our proposed neural predictors to appear the most in Table 18.

Finding 7. From amongst the three variations of our proposed neural predictors, the Neural Aggregated Matching (NAM) feature, which considers the maximum similarity obtained from the neural representations of the tokens in the table and query spaces, has the best overall performance.

Furthermore, as neural embeddings play a central role in our proposed neural predictors, it is important to also understand whether the choice of neural embedding representations impact the performance of our neural predictors. There have recently been several variations of embeddings such as BERT [66] and ELMo [67], that have shown impressive performance for tasks such as

Table 19: Impact of embedding type on the performance of P_{WIG}^{NAM} for each ranking function in terms of Kendall's correlation. Bold values show the largest value per ranking function.

$NDCG$@20 Kendall	Ranking Function						
	Content-based				Feature-based		
	STR	Single Field	Table2VecW	Table2VecE	WikiTable	WebTable	LTR
Word2vec	0.234	0.28	**0.26**	0.264	**0.2**	0.019	0.212
BERT	**0.268**	0.262	0.208	**0.289**	0.19	**0.388**	0.23
ELMo	0.246	**0.296**	0.25	0.246	0.157	0.231	**0.25**

question answering and language understanding. For the sake of comparison, we compare the performance of the best interpolated variation of our neural predictor, i.e., P_{WIG}^{NAM}, for cases when Word2vec-based [20] Google News word embeddings are adopted to when embeddings from BERT and ELMo are used. The results are shown in Table 19.

As seen in the table,the obtained results do not show one particular type of embedding outperforming the other by a significant margin. While the Word2vec embeddings show favorable performance on the Table2VecW and WikiTable, it is BERT that shows a better performance on STR, Table2VecE, and WebTable. Also, ELMo shows the best performance on Single Field and LTR ranking functions We note that except for WebTable where the performance of BERT and ELMo is substantially better than word2vec, the results on other ranking functions are comparable.

Finding 8. The performance of the best interpolated variation of our neural predictor when customized based on either Word2vec, BERT or ELMo neural embeddings does not seem to depend on the type of the ranking function and in principal one embedding type does not outperform the other (except by BERT on WebTable).

4.5.4. RQ4. Impact of Table Characteristics on the Neural Predictors

The objective the last research question is to understand whether the consideration of table-specific characteristics such as table rows, columns, and cells have any impact on the quality of query performance prediction. In the previous research questions, we have considered each table to be a single document and hence all the content in the table cells were concatenated for the sake of query performance prediction. In this research question, we investigate whether the consideration

Table 20: The impact of table characteristics on the performance of P_{WIG}^{NAM} for each ranking function in terms of Kendall's correlation. Bold values show the largest value per ranking function.

$NDCG$@20 Kendall	Ranking Function						
	Content-based				Feature-based		
	STR	Single Field	Table2VecW	Table2VecE	WikiTable	WebTable	LTR
Row-based	0.188	0.194	0.158	0.114	0.028	**0.118**	0.095
Column-based	0.183	0.202	0.165	0.092	0.07	0.081	0.108
Cell-based	0.157	0.165	0.169	0.097	0.021	0.05	0.069
Unstructured	**0.234**	**0.28**	**0.26**	**0.264**	**0.2**	0.019	**0.212**

of each of these table elements leads to any impact on QPP performance.

In order to consider table elements for performing QPP, we develop embedding representations for each table element, i.e., cell, column or row, by taking the average of the embeddings of the content observed in that element. For instance, the representation for a table cell is the average of the embedding representations of the content in that cell. We adopt these table element representations to compute our proposed neural predictors by averaging over table element embeddings. Without loss of generality, we report our findings on the best interpolated variation of our neural predictor, i.e., P_{WIG}^{NAM} in Table 20. As seen in the table, the last row is the situation where the whole table is considered to be a single document. We find that the consideration of the whole table as a single document produces the best performance for the QPP task on all ranking functions except for WebTable. This seems to be inline with the findings reported by Deng et al [17] who found that the consideration of a table as a document leads the improved retrieval performance for the ad hoc table retrieval task as well.

Finding 9. While table-specific characteristics such as rows, columns and cells provide richer content presentation and better interpretability, they do not necessarily lead to improved QPP performance. The consideration of the whole table as a document shows the best QPP performance over all ranking functions (except WebTable).

5. Concluding Remarks

The main objective of our work in this paper has been to explore how neural embedding techniques can be exploited to predict query performance in the ad hoc table retrieval task. Our motivation is that neural embedding techniques enable us to estimate the performance of a table ranking function based on the semantic content of the query and the retrieved table list. Neural embedding techniques assign a low-dimensional vector to each token in a new space such that the properties of, and the relationships between, the tokens are preserved. In this space, the vectors of tokens that are semantically or syntactically similar tend to be close.

We proposed a set of neural predictors that represent the content of tables and queries as bag of tokens, which are then systematically integrated into existing state-of-the-art post-retrieval query-performance predictors. The proposed neural features employ neural embeddings to capture the semantic relationship between the representation of tables and queries based on their relationship in the embedding space.

We evaluated the effectiveness of the neural predictor through a complete set of experiments on a test collection that consists of 1.6M WikiTables and 60 queries. We compared the ranking of queries by the neural predictor and actual performance measured according to their true NDCG using Pearson's correlation to assess the neural predictor quality. We summarize our key findings from the results of the experiments as follows:

- A bag of words representation is the more suitable representation for tables and queries in the context of QPP for ad hoc table retrieval. A bag of entity representation does not show equally effective performance especially because of the sparsity of table entity representations;
- Our proposed neural predictors are effective in improving the performance of existing QPP methods when predicting query difficulty for content-based ranking function. The proposed predictors do not contribute to better QPP performance for feature-based ranking functions;
- From among the proposed neural features, NAM, which captures the maximum similarity between the query and table tokens in the embedding space, leads to a more accurate estimation of query performance compared to other proposed neural features.

Finally, we would like to highlight that as this work is among the very first attempts to perform query performance prediction for ad hoc table retrieval, the prediction performances would not be

considered suitable to be used in production. Our work shows that while the consideration of neural predictors do provide better performance prediction results compared to existing state of the art QPP methods, the absolute correlation values are still low (for both our approach and existing QPP methods in the literature). As such, we believe there is optimistically room for much better work to build on the foundations of our work.

References

[1] F. Diaz, "Pseudo-query reformulation," in *European Conference on Information Retrieval*, pp. 521–532, Springer, 2016.

[2] Y. Zhou and W. B. Croft, "Query performance prediction in web search environments," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 543–550, 2007.

[3] A. Shtok, O. Kurland, D. Carmel, F. Raiber, and G. Markovits, "Predicting query performance by query-drift estimation," *ACM Transactions on Information Systems (TOIS)*, vol. 30, no. 2, pp. 1–35, 2012.

[4] N. Arabzadeh, F. Zarrinkalam, J. Jovanovic, F. Al-Obeidat, and E. Bagheri, "Neural embedding-based specificity metrics for pre-retrieval query performance prediction," *Information Processing & Management*, vol. 57, no. 4, p. 102248, 2020.

[5] D. Carmel and O. Kurland, "Query performance prediction for ir," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 1196–1197, 2012.

[6] B. He and I. Ounis, "Inferring query performance using pre-retrieval predictors," in *International symposium on string processing and information retrieval*, pp. 43–54, Springer, 2004.

[7] S. Tomlinson, "Robust, web and terabyte retrieval with hummingbird searchservertm at trec 2004," in *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004*, National Institute of Standards and Technology (NIST), 2004.

[8] G. Zuccon, B. Koopman, P. Bruza, and L. Azzopardi, "Integrating and evaluating neural word embeddings in information retrieval," in *Proceedings of the 20th Australasian document computing symposium*, pp. 1–8, 2015.

[9] B. Mitra, N. Craswell, *et al.*, "An introduction to neural information retrieval," *Foundations and Trends® in Information Retrieval*, vol. 13, no. 1, pp. 1–126, 2018.

[10] E. Bagheri, F. Ensan, and F. Al-Obeidat, "Neural word and entity embeddings for ad hoc retrieval," *Information Processing & Management*, vol. 54, no. 4, pp. 657–673, 2018.

[11] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pp. 101–110, 2014.

[12] G. W. Furnas, S. Deerwester, S. T. Durnais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum, "Information retrieval using a singular value decomposition model of latent semantic structure," in *ACM SIGIR Forum*, vol. 51, pp. 90–105, ACM New York, NY, USA, 2017.

[13] F. Ensan and E. Bagheri, "Document retrieval model through semantic linking," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, pp. 181–190, 2017.

[14] S. Zhang and K. Balog, "Ad hoc table retrieval using semantic similarity," in *Proceedings of the 2018 World Wide Web Conference*, pp. 1553–1562, 2018.

[15] E. Bagheri and F. Al-Obeidat, "A latent model for ad hoc table retrieval," in *Advances in Information Retrieval*, pp. 86–93, Springer International Publishing, 2020.

[16] A. L. Gentile, P. Ristoski, S. Eckel, D. Ritze, and H. Paulheim, "Entity matching on web tables: a table embeddings approach for blocking," in *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017* (V. Markl, S. Orlando, B. Mitschang, P. Andritsos, K. Sattler, and S. Breß, eds.), pp. 510–513, OpenProceedings.org, 2017.

[17] L. Deng, S. Zhang, and K. Balog, "Table2vec: Neural word and entity embeddings for table population and retrieval," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, (New York, NY, USA), p. 1029–1032, Association for Computing Machinery, 2019.

[18] S. Zhang and K. Balog, "Web table extraction, retrieval, and augmentation: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 2, pp. 13:1–13:35, 2020.

[19] J. Herzig, P. K. Nowak, T. Muller, F. Piccinno, and J. Eisenschlos, "Tapas: Weakly supervised table parsing via pre-training," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4320–4333, 2020.

[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of Workshop at ICLR*, 2013.

[22] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[23] C. S. Bhagavatula, T. Noraset, and D. Downey, "Methods for exploring and mining tables on wikipedia," in *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, pp. 18–26, 2013.

[24] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang, "Webtables: exploring the power of tables on the web," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 538–549, 2008.

[25] R. Pimplikar and S. Sarawagi, "Answering table queries on the web using column keywords," in *Proceedings of the VLDB Endowment*, pp. 908–919, 2012.

[26] S. Zhang and K. Balog, "Auto-completion for data cells in relational tables," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 761–770, 2019.

[27] S. Zhang and K. Balog, "On-the-fly table generation," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 595–604, 2018.

[28] N. Tax, S. Bockting, and D. Hiemstra, "A cross-benchmark comparison of 87 learning to rank methods," *Information processing & management*, vol. 51, no. 6, pp. 757–772, 2015.

[29] G. Capannini, C. Lucchese, F. M. Nardini, S. Orlando, R. Perego, and N. Tonellotto, "Quality versus efficiency in document scoring with learning-to-rank models," *Information Processing & Management*, vol. 52, no. 6, pp. 1161–1177, 2016.

[30] D. Metzler and W. B. Croft, "Combining the language model and inference network approaches to retrieval," *Information processing & management*, vol. 40, no. 5, pp. 735–750, 2004.

[31] G. Limaye, S. Sarawagi, and S. Chakrabarti, "Annotating and searching web tables using entities, types and relationships," *Proc. VLDB Endow.*, vol. 3, p. 1338–1347, Sept. 2010.

[32] A. Das Sarma, L. Fang, N. Gupta, A. Halevy, H. Lee, F. Wu, R. Xin, and C. Yu, "Finding related tables," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pp. 817–828, 2012.

[33] T. T. Nguyen, Q. V. H. Nguyen, M. Weidlich, and K. Aberer, "Result selection and summarization for web table search," in *2015 IEEE 31st International Conference on Data Engineering*, pp. 231–242, IEEE, 2015.

[34] M. Yakout, K. Ganjam, K. Chakrabarti, and S. Chaudhuri, "Infogather: Entity augmentation and attribute discovery by holistic matching with web tables," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, (New York, NY, USA), p. 97–108, Association for Computing Machinery, 2012.

[35] M. Zhang and K. Chakrabarti, "Infogather+: Semantic matching and annotation of numeric and time-varying attributes in web tables," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, (New York, NY, USA), p. 145–156, Association for Computing Machinery, 2013.

[36] S. Tyree, K. Q. Weinberger, K. Agrawal, and J. Paykin, "Parallel boosted regression trees for web search ranking," in *Proceedings of the 20th international conference on World wide web*, pp. 387–396, 2011.

[37] D. Carmel and E. Yom-Tov, "Estimating the query difficulty for information retrieval," *Synthesis Lectures on Information Concepts, Retrieval, and Services*, vol. 2, no. 1, pp. 1–89, 2010.

[38] R. Cummins, M. Lalmas, C. O'Riordan, and J. M. Jose, "Navigating the user query space," in *International Symposium on String Processing and Information Retrieval*, pp. 380–385, Springer, 2011.

[39] S. Déjean, R. T. Ionescu, J. Mothe, and M. Z. Ullah, "Forward and backward feature selection for query performance prediction," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pp. 690–697, 2020.

[40] S. Cronen-Townsend, Y. Zhou, and W. B. Croft, "Predicting query performance," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 299–306, 2002.

[41] Y. Zhou and W. B. Croft, "Ranking robustness: a novel framework to predict query performance," in *Proceedings of the 15th ACM international conference on Information and knowledge management*, pp. 567–574, 2006.

[42] J. A. Aslam and V. Pavlu, "Query hardness estimation using jensen-shannon divergence among multiple scoring functions," in *European conference on information retrieval*, pp. 198–209, Springer, 2007.

[43] D. Metzler and W. B. Croft, "A markov random field model for term dependencies," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 472–479, 2005.

[44] Y. Tao and S. Wu, "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, 2014.

[45] J. A. Shaw and E. A. Fox, "Combination of multiple searches," in *Proceedings of The Third Text REtrieval Conference, TREC 1994*, pp. 105–108, National Institute of Standards and Technology (NIST), 1994.

[46] G. Markovits, A. Shtok, O. Kurland, and D. Carmel, "Predicting query performance for fusion-based retrieval," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 813–822, 2012.

[47] H. Roitman, "Enhanced performance prediction of fusion-based retrieval," in *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*, pp. 195–198, 2018.

[48] H. Zamani, W. B. Croft, and J. S. Culpepper, "Neural query performance prediction using weak supervision from multiple signals," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 105–114, 2018.

[49] R. Cummins, J. Jose, and C. O'Riordan, "Improved query performance prediction using standard deviation," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 1089–1090, 2011.

[50] B. Siddiquie, R. S. Feris, and L. S. Davis, "Image ranking and retrieval based on multi-attribute queries," in *CVPR 2011*, pp. 801–808, IEEE, 2011.

[51] C. S. Bhagavatula, T. Noraset, and D. Downey, "Tabel: entity linking in web tables," in *International Semantic Web Conference*, pp. 425–441, Springer, 2015.

[52] S. Zhang and K. Balog, "Entitables: Smart assistance for entity-focused tables," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 255–264, 2017.

[53] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, pp. 1188–1196, 2014.

[54] E. Nalisnick, B. Mitra, N. Craswell, and R. Caruana, "Improving document ranking with dual word embeddings," in *Proceedings of the 25th International Conference Companion on World Wide Web*, pp. 83–84, 2016.

[55] I. Vulić and M.-F. Moens, "Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 363–372, 2015.

[56] T. Kenter and M. De Rijke, "Short text similarity with word embeddings," in *Proceedings of the 24th ACM international on conference on information and knowledge management*, pp. 1411–1420, 2015.

[57] L. Galke, A. Saleh, and A. Scherp, "Word embeddings for practical information retrieval," in *INFORMATIK 2017*, pp. 2155–2167, Gesellschaft für Informatik, Bonn, 2017.

[58] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pp. 59–66, IEEE, 1998.

[59] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *International conference on machine learning*, pp. 957–966, 2015.

[60] M. J. Cafarella, A. Halevy, and N. Khoussainova, "Data integration for the relational web," *Proceedings of the*

VLDB Endowment, vol. 2, no. 1, pp. 1090–1101, 2009.

[61] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[62] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[63] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.

[64] Y. Li, R. Zheng, T. Tian, Z. Hu, R. Iyer, and K. Sycara, "Joint embedding of hierarchical categories and entities for concept categorization and dataless classification," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2678–2688, 2016.

[65] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *COLING 2018, 27th International Conference on Computational Linguistics*, pp. 1638–1649, 2018.

[66] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), pp. 4171–4186, Association for Computational Linguistics, 2019.

[67] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)* (M. A. Walker, H. Ji, and A. Stent, eds.), pp. 2227–2237, Association for Computational Linguistics, 2018.