Information Processing and Management 57 (2020) 102248

Contents lists available at ScienceDirect

Information Processing and Management

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

Neural embedding-based specificity metrics for pre-retrieval query performance prediction

Negar Arabzadeh[a], Fattane Zarrinkalam[a], Jelena Jovanovic[b], Feras Al-Obeidat[c], Ebrahim Bagheri*,[a]

[a] Electrical and Computer Engineering, Ryerson University, 245 Church Street, Canada
[b] University of Belgrade, Serbia
[c] Zayed University, United Arab Emirates

ARTICLE INFO

Keywords:
Performance prediction
Neural embeddings
Ad hoc retrieval

ABSTRACT

In information retrieval, the task of query performance prediction (QPP) is concerned with determining in advance the performance of a given query within the context of a retrieval model. QPP has an important role in ensuring proper handling of queries with varying levels of difficulty. Based on the extant literature, *query specificity* is an important indicator of query performance and is typically estimated using corpus-specific frequency-based specificity metrics However, such metrics do not consider term semantics and inter-term associations. Our work presented in this paper distinguishes itself by proposing a host of corpus-independent specificity metrics that are based on pre-trained neural embeddings and leverage geometric relations between terms in the embedding space in order to capture the semantics of terms and their interdependencies. Specifically, we propose three classes of specificity metrics based on pre-trained neural embeddings: neighborhood-based, graph-based, and cluster-based metrics. Through two extensive and complementary sets of experiments, we show that the proposed specificity metrics (1) are suitable specificity indicators, based on the gold standards derived from knowledge hierarchies (Wikipedia category hierarchy and DMOZ taxonomy), and (2) have better or competitive performance compared to the state of the art QPP metrics, based on both TREC ad hoc collections namely Robust'04, Gov2 and ClueWeb'09 and ANTIQUE question answering collection. The proposed graph-based specificity metrics, especially those that capture a larger number of inter-term associations, proved to be the most effective in both query specificity estimation and QPP. We have also publicly released two test collections (i.e. specificity gold standards) that we built from the Wikipedia and DMOZ knowledge hierarchies.

1. Introduction

The problem of predicting an information retrieval system's performance for a given query is called *Query Performance Prediction (QPP)*. QPP methods can be broadly categorized into two groups: *Pre-retrieval QPP* and *Post-retrieval QPP*. The former predicts the retrieval performance of a given query without having access to the retrieved documents for that query while the latter uses the query and the set of retrieved documents for the query to predict retrieval performance. Although post-retrieval predictors outperform pre-retrieval predictors, they are more expensive in terms of computation and time complexity. Pre-retrieval query performance predictors can be classified into (1) similarity-based (2) coherency-based (3) term relatedness-based, and (4) specificity-based methods

* Corresponding author.

https://doi.org/10.1016/j.ipm.2020.102248
Received 7 November 2019; Received in revised form 6 February 2020; Accepted 13 March 2020
Available online 08 April 2020

(Hauff, Hiemstra, & de Jong, 2008). The focus of this paper is on proposing novel *specificity-based pre-retrieval QPP methods*.

Specificity-based pre-retrieval QPP methods are based on estimating the specificity of terms for addressing pre-retrieval QPP (Thomas, Scholer, Bailey, & Moffat, 2017). Specificity has traditionally been estimated using corpus-specific frequency statistics (Jones, 2004). While being widely adopted in the literature, frequency-based specificity metrics can suffer from some shortcomings including the lack of attention to the semantics of terms and their inter-dependencies. Our objective in this paper is to overcome such deficiencies and define a host of specificity metrics that consider *term semantics* and *inter-term associations*. To that end, we consider neural embedding-based representation of terms since such a representation allows for measuring semantic relatedness (i.e., associations) among the terms (Mikolov, Chen, Corrado, & Dean, 2013a).

Given the fact that neural embeddings preserve geometric properties of term associations (Mimno & Thompson, 2017), they can be helpful for defining term specificity. We define neural embedding-based measures of specificity and use them for performing QPP. By using pre-trained neural embeddings, unlike existing pre-retrieval QPP methods (Hauff, 2010; He, Larson, & de Rijke, 2008; Zhao, Scholer, & Tsegay, 2008), our work is not dependent on collection-specific frequency statistics. Therefore, performance prediction can be computed even for query terms that are not frequent in the document collection. For instance, predicting the performance of queries containing less frequently observed terms in the document collection is a major challenge for frequency-based metrics. In our proposed approach, such terms can be handled using their relative position to other terms within the embedding space.

Our work is driven by the intuition that, in a neural-embedding space, more specific terms have a higher likelihood of being surrounded by a higher number of terms compared to generic terms. In other words, given the fact that highly specific terms express precise semantics, they are more likely to be highly related to other terms (and hence be surrounded by more terms) whereas generic terms tend to have weaker relationships with other terms (Jones, 2004). Based on this idea, we derive and define three groups of specificity metrics based on their characteristics, including (1) Neighborhood-based metrics, (2) Graph-based metrics, and (3) Cluster-based metrics.

Additionally, we introduce and publicly share two test collections to evaluate the specificity metrics, which are structured collections of terms with associated human-defined specificity values. The test collections have been derived from the Wikipedia category hierarchy, and the DMOZ taxonomy with the understanding that categories higher up in the hierarchy are more generic, and those further down in the hierarchy are more specific.

In addition, we evaluate the performance of the proposed specificity-based pre-retrieval QPP methods based on three widely used TREC corpora, namely Robust04, ClueWeb09, and Gov2, and ANTIQUE question answering collection and their corresponding topic sets over TREC benchmarks (Voorhees, 2005a; 2005b). Compared to the existing pre-retrieval QPP methods, our experiments show that our proposed neural embedding-based specificity metrics are effective in predicting query performance. Furthermore, our work distinguishes itself from existing frequency-based pre-retrieval metrics by (1) preserving the semantic aspects of terms and dependencies among them, and (2) proposing corpus-independent predictors by utilizing pre-trained neural embeddings. More specifically, the key contributions of our work are as follows:

1. We formally introduce the task of predicting corpus-independent term specificity based on a collection of neural embeddings.
2. We propose a host of unsupervised specificity metrics for predicting specificity based on neural embeddings. Further, we show that our proposed neural embedding-based metrics are robust to the choice of the pre-trained embeddings by assessing them over three different pre-trained embeddings.
3. We show how the proposed neural embedding-based specificity metrics can serve as pre-retrieval QPP methods. The proposed specificity metrics show promising performance when predicting query performance.
4. We offer two gold standard test collections consisting of term specificity measurements defined in relation to Wikipedia and DMOZ categories. These collections are made publicly available for replication studies.

In the next section, we provide some information on background of pre-retrieval QPP and related work on specificity application. The proposed neural embedding-based specificity metrics are introduced in Section 3. Section 4 focuses on the evaluations of the proposed metrics directly using two gold standard test collections, followed by evaluating our proposed specificity-based pre-retrieval QPP methods. Section 6 is dedicated to a discussion about our research findings. Finally, Section 7 sheds light on future work and concludes the paper.

2. Related work

The objective of this paper is to propose a set of specificity metrics for pre-retrieval QPP. Therefore, in this section, we first provide some background information on different types of pre-retrieval QPP methods. Then, we review different studies which have proposed or incorporated *specificity* metrics in their models.

2.1. Pre-retrieval QPP methods

Several pre-retrieval QPP methods are based on *linguistic* and *statistical* predictors (Carmel & Yom-Tov, 2010). While linguistic-based predictors use natural language processing methods, such as the number of morphs per query or the average number of synonyms per query, to analyze a query, they do not offer competitive performance in predicting the retrieval system's performance (Mothe & Tanguy, 2005).

Statistical pre-retrieval QPP methods are based on statistical properties such as *Similarity, Coherency, Relatedness* or *Specificity*

metrics (Carmel & Yom-Tov, 2010). (1) Similarity-based pre-retrieval QPP methods, e.g., *Collection Query similarity (SCQ)* (Zhao et al., 2008), measure the quality of the retrieved documents based on the similarity between the query and the corpus of documents. If a query and a corpus are highly similar, most probably multiple documents exist in the corpus that correspond to the query. (2) Coherency-based pre-retrieval QPP methods, e.g., *Clarity Score (CS)* (He et al., 2008) and *term weight Variance (VAR)* (Zhao et al., 2008), calculate the inter-similarity of the retrieved documents which contain the query terms. (3) Relatedness-based pre-retrieval QPP methods, e.g., *Point-wise Mutual Information (PMI)* (Hauff, 2010), evaluate the co-occurrence of query terms in a corpus, through which more frequent co-occurring query terms indicate that the query is less difficult to answer. (4) Specificity-based pre-retrieval QPP methods calculate the specificity of query terms based on their distribution over the corpus: the more specific the query terms are, the easier they will be.

Since our proposed approach can be classified as a specificity-based approach, in the following, we review existing pre-retrieval specificity-based QPP methods. For example, *Inverse Document Frequency (IDF)* and *Inverse Collection Term Frequency (ICTF)* (Kwok, 1996) are two metrics to estimate query term specificity for query performance prediction based on the idea that the higher the average value of these metrics over the query terms are, the easier it is to satisfy the query. The *Query Scope (QS)* predictor (He & Ounis, 2004) is another state-of-the-art specificity-based pre-retrieval QPP method, which determines the percentage of documents in the collection that include at least one of the query terms. A higher query scope signifies that there are multiple documents pertaining to the query, however it may be difficult to distinguish between the relevant and non-relevant results. Further, the *Simplified Clarity Score (SCS)* predictor (He & Ounis, 2004) considers the length of the query while measuring the specificity of the query by determining the divergence between the simplified query language model and the collection language model.

Recently, a specificity metric known as $P_{clarity}$ is proposed (Roy, Ganguly, Mitra, & Jones, 2019) that utilizes the neural embedding-based representation of query terms to predict query performance. Similar to our cluster-based specificity metrics, $P_{clarity}$ is based on the idea that the higher probability of possible clusters around a term in an embedding space indicates higher generality of the term. Whereas our cluster-based metrics consider all the term-clusters around the ego term uniformly, they have focused only on the characteristics of the dominant cluster. In other words, high variance between the number of elements in the clusters around a term could serve as a potential indication for specificity.

2.2. Applications for term specificity

Term specificity is used in several applications beyond QPP, especially within the context of the Web, including user interest modeling, and social media personalization, among others. For example, Orlandi, Kapanipathi, Sheth, and Passant (2013) have utilized term specificity in order to recommend more relevant entities to users on the social web. They claimed that while their proposed specificity metrics can be calculated and updated in real-time, they are also domain-independent. They employed the Linked Data graph and analyzed the predicates connecting entities in the graph. They considered the relation between incoming and outgoing predicates as a gauge for specificity. Although our work is different from Orlandi et al. (2013) in multiple ways, using a graph to measure the specificity and preserving semantic aspects of terms is what our work has in common with Orlandi et al. (2013). Their proposed metric is defined on a directed graph whose nodes are entities. However, we structure our network as an undirected graph whose nodes are the embedding vectors. In our work, the appropriate indicator of specificity is the dispersion of embedding vectors in the embedding space which preserve the semantic aspects of the terms, while in Orlandi et al. (2013), the number of unique predictors among entities is the measure for specificity.

The concept of specificity has also been utilized for social media analysis. For example, Benz, Körner, Hotho, Stumme, and Strohmaier (2011) leveraged specificity to measure tag similarity and tag relatedness in social metadata. The authors proposed three generality measures including frequency-based, entropy-based and centrality-based measures. Our work is similar to theirs in the sense that we both estimate specificity based on the centrality degrees on a graph. However, the idea behind building the underlying term graphs are totally different. Since Benz et al. (2011) proposed the specificity/generality metrics to use them in social media applications, the authors define a simple term graph where there exists an edge between two nodes (terms) if there is at least one post containing both of them. However, we build a weighted graph in an embedding space in which the terms are connected to each other if they have a similarity higher than a specific threshold. Therefore, our work benefits from preserving the semantic aspects of the terms. We both show that measuring betweenness centrality, closeness centrality and degree centrality in a graph can be an appropriate indicators for specificity.

Direct evaluation of specificity metrics is a challenging task since specificity can be subjective and biased by personal experiences (Orlandi et al., 2013). Therefore, we cannot have an exact value denoting term specificity and consequently, no absolute ground truth exists for evaluating specificity. One of the common ways of building a gold standard for evaluating specificity is to use the human's perception by either asking a number of people to rate different terms' specificity/generality or to classify the terms into specific and generic classes (Orlandi et al., 2013). However, generating such a gold standard might be inefficient, expensive, and impractical in practice.

On the other hand, well-established term hierarchies reflect a fair agreement with human judgments (Benz et al., 2011). Since the level of each term in a hierarchy can be a suitable indicator of the term's specificity, taxonomical or ontological hierarchies can be used as gold standards in order to evaluate different levels of specificity (Benz et al., 2011; Kammann & Streeter, 1971; Orlandi et al., 2013). For example, Benz et al. (2011) studied different levels of generality by identifying the hierarchical relationships between terms. They leveraged several taxonomies and ontologies such as WordNet, Yago, DMOZ, and WikiTaxonomy as the gold standard for specificity. In addition, Orlandi et al. (2013) utilized the DMOZ taxonomy as one of the baselines to assess their proposed specificity method. Based on the same idea, we utilize the position of a term in the Wikipedia Category Hierarchy and the DMOZ taxonomy as

the gold standard for estimating term specificity.

In our previous work (Arabzadeh, Zarrinkalam, Jovanovic, & Bagheri, 2019), we proposed a set of specificity metrics based on neural embeddings of the terms that consider term semantics and inter-term associations. The current paper extends our previous work with several improvements: (1) we propose a larger number of specificity metrics; (2) we provide a more comprehensive analysis and review of the related work (3) we provide a new test collection based on DMOZ taxonomy, (4) we utilize our proposed neural embedding-based specificity metrics in the context of pre-retrieval query performance prediction, and (5) finally, more comprehensive experiments are conducted and new findings are reported.

3. The proposed approach

In this section, we propose some metrics that utilize the neural embedding representation of terms to estimate the specificity of terms to predict query performance. In the following, we first provide some preliminary information required to define our specificity metrics, then in Section 3.2, we introduce the proposed specificity metrics which are divided into three categories based on their characteristics, including (1) Neighborhood-based, (2) Graph-based and (3) Cluster-based metrics.

3.1. Preliminaries

Our work focuses on how vector representations of terms within a given embedding space, i.e., geometric properties of neural embeddings, can be used to define appropriate metrics for estimating term specificity. Our intuition is that the local neighborhood of a term in a given embedding space can be used to derive indicators of the term's specificity. It is based on the fact that, in embedding space, two semantically related terms have similar embedding vectors (as measured, e.g., by cosine similarity of their vectors) (Mikolov et al., 2013a).

We select the local neighborhood of a term t_i in an embedding space, by retrieving a set of highly similar terms to t_i. Formally, let $\mu(t_i)$ be the degree of similarity of the most similar term to t_i in the embedding space. We select the $\varepsilon - neighborhood$ of t_i, denoted as $N_\varepsilon(t_i)$, as Eq. (1).

$$N_\varepsilon(t_i) = \{t_j: \frac{\upsilon_{t_i}.\ \upsilon_{t_j}}{\|\upsilon_{t_i}\|\|\upsilon_{t_j}\|} \geqslant \varepsilon \times \mu(t_i)\} \tag{1}$$

Simply put, for a given term t, we calculate the cosine similarity between its embedding vector and other terms' embedding vectors in the embedding space and the terms with a semantic relatedness higher than $\varepsilon \times \mu(t)$, are selected as the $\varepsilon - neighborhood$ of term t. The $\varepsilon - neighborhood$ of two terms (e.g. Technology and iPhone) are illustrated in Fig. 1. For instance, assuming 'iPhone' is the ego term and $\varepsilon = 0.85$, given 'ios_apple' term is the most similar one to the ego with a semantic relatedness of 0.90, its $\varepsilon - neighborhood$ will consist of all the terms in the embedding space such as 'smartphonoes', 'ipad' and 'ipod', that have a semantic relatedness above 0.85×0.90, i.e. 0.765 to 'iPhone'.

Next, in order to examine inter-term associations in the neighborhood of t, we define the notion of an $ego - network$ in Definition 1.

Definition 1. Ego-network: An ego-network for term t_i, denoted as $\xi(t_i) = (\mathbb{V}_{ego}, \mathbb{E}_{ego}, g)$, is a weighted undirected graph where $\mathbb{V}_{ego} = \{t_i\} \cup N_\varepsilon(t_i)$, and $\mathbb{E}_{ego} = \{e_{t_i,t_j}: \forall\ t_i, t_j \in \mathbb{V}_{ego}\}$. The function $g: \mathbb{E}_{ego} \rightarrow [0, 1]$ is the cosine semantic relatedness between the embedding vectors of two incident terms of an edge e_{t_i,t_j}, i.e., v_{t_i} and v_{t_j}. We refine $\xi(t_i)$ by pruning any edge with a weight below $\varepsilon \times \mu(t_i)$.

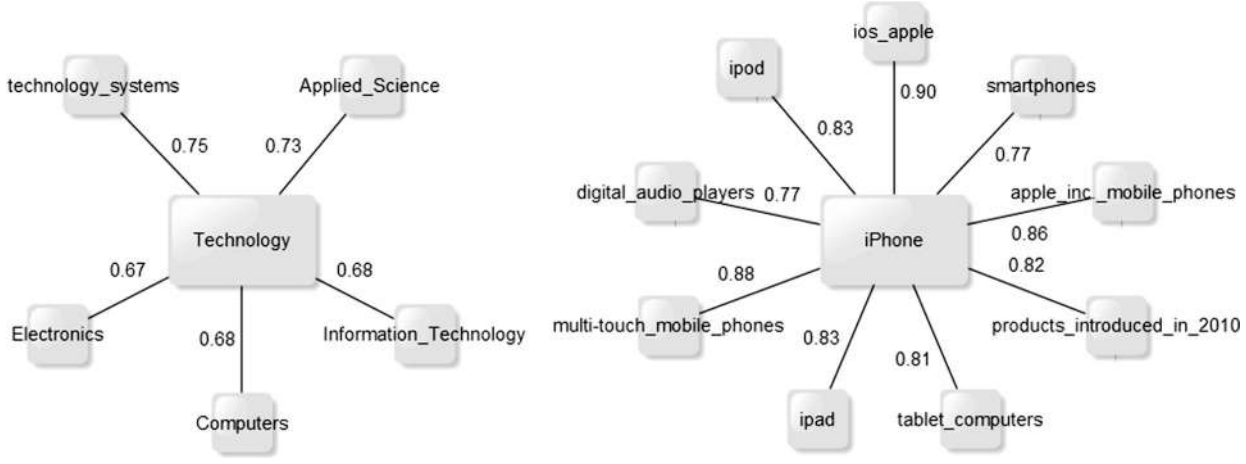

Fig. 1. ε-neighborhood for terms Technology and iPhone.

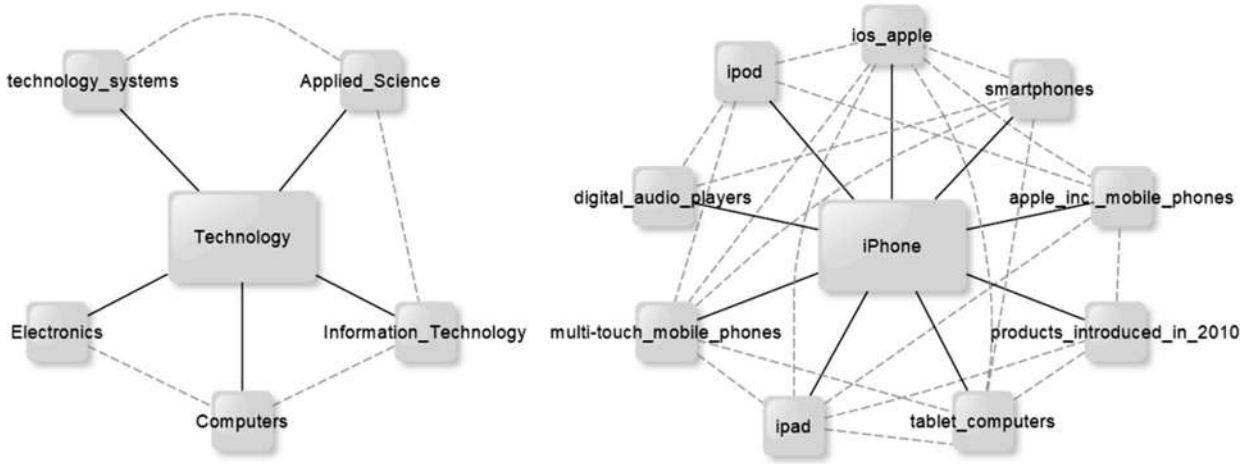

Fig. 2. A subset of the ego-networks for terms Technology and iPhone. It should be noted that the edge weights are not presented to reduce clutter.

In the above definition, we propose to build an $ego-network$ for term t_i such that t_i is the ego node and is connected directly to other terms only if the degree of semantic relatedness between the ego and its neighbors is above a given threshold. Simply put, any two nodes in the $\varepsilon-neighborhood$ of the term t_i with a semantic relatedness to t_i higher than $\varepsilon \times \mu(t_i)$, are connected to each other in the $ego-network$ of t_i. Fig. 2 shows the ego networks for the specific term 'iPhone' and for the generic term 'Technology'. As shown in Fig. 2, for example, the immediate neighbors of the 'iPhone' include terms such as 'iPod', 'smartphones', 'tablet_computers', among others. In this $ego-network$, the neighbors are also connected to each other if their semantic relatedness is more than 0.765, derived based on $\varepsilon \times \mu(t_i)$.

3.2. Specificity metrics

In light of the presented preliminaries, in this section, we introduce our proposed specificity metrics in three categories, namely Neighborhood-based, Graph-based and Cluster-based.

3.2.1. Neighborhood-based

Neighborhood-based metrics only consider connections between the ego node and its immediate neighbors. The intuition behind this category of metrics is that (1) as highly specific terms express precise semantics, they have a high likelihood of being surrounded, in the embedding space, by a higher number of terms compared to generic terms. (2) The semantic relatedness between terms would be notably weaker than in the case of specific terms' local neighborhood, which are highly semantically related to one another. Therefore, by computing and considering the connections of a term within its local neighborhood, it is possible to differentiate between specific and generic terms. For example, the specific term 'iPhone' is highly similar to the terms referring to specific devices such as 'iPod' and 'iPad'. However, since a generic term, e.g., 'technology', is often related to many different terms with diverse senses, such as 'Applied_sciences' and 'Computers', it would end up having weaker relationships with these diverse neighbors.

Therefore, Based on this idea, we define neighborhood-based metrics summarized in Table 1 and explained in the following.

We define *Neighborhood Size (NS)* metric to measure the specificity of a term based on the idea that given a constant semantic relatedness threshold, specific terms are surrounded by more similar terms compared to generic terms. For example, in Fig. 1, assuming ε is 0.85, by comparing the $\varepsilon-neighborhood$ of the terms 'Technology' and 'iPhone', one can observe that, neighborhood size (NS) of 'Technology' which is a generic term is less than the Neighborhood size of 'iPhone' as a specific term. In general, we hypothesize that the size of the $\varepsilon-neighborhood$ of a term t relates directly to the specificity of t.

In network theory, the centrality of a node in a network is usually a hint to the importance of the node. Having said that, centrality

Table 1
The set of proposed neighborhood-based specificity metrics. $V_N(t)$ denotes the sum of the embedding vectors of the terms in the $\varepsilon-neighborhood$ of term t.

Metric name	Description
Neighborhood Size (NS)	The number of terms in ε-neighborhood of term t
Weighted Degree Centrality (WDC)	The weighted degree centrality of the ego node in the ε-neighborhood graph
Median Absolute Deviation (MAD)	The Median Absolute Deviation of weight of edges directly connected to t
Neighborhood Variance (NV)	The variance of weight of edges directly connected to t
Most Similar Neighbor (MSN)	The maximum weight of edges directly connected to t
Neighborhood Vector Similarity (NVS)	semantic relatedness of t to $V_N(i)$
Neighborhood Vector Magnitude (NVM)	Magnitude of $V_N(t)$

can be mulled over as a degree of abstractness since more unique terms are more potent. Degree centrality is the simplest among the measures of node centrality, as it relies merely on the number of the node's immediate neighbors. In a weighted network, degree centrality has generally been extended to the sum of weights of immediate neighbors. Therefore, by taking $\varepsilon - neighborhood$ as a network, calculating *Weighted Degree Centrality (WDC)* of a term t in this network is expected to lead to a possible measurement of its specificity since weights reflect the similarity of the neighboring terms. For instance, in Fig. 1, the WDC of the specific term 'iPhone' is 7.47 which is greater than the WDC of the generic term 'Technology' that is equal to 3.51. Hence, we hypothesize that higher WDC can be interpreted as higher specificity.

As another neighborhood-based specificity metric, we define *Median Absolute Deviation (MAD)*. Based on the assumption that the semantic relatedness in a neighborhood of a specific term is less dispersed compared to a generic term, this metric calculates how far each data point is from its median. Similarly, we define the variance of the semantic relatedness of neighbors of a term, i.e. *Neighborhood Variance (NV)* as a specificity metric.

Moreover, we define *Most Similar Neighbor (MSN)* metric based on the intuition that the semantic relatedness between a term and its most similar neighbour is highly related to how specific the term is. As it is shown in Fig. 1, the most similar neighbor to the specific term 'iPhone' is 'ios_apple' with the semantic relatedness of 0.90 which is greater than the degree of semantic relatedness of the most similar neighbor to the generic term 'Technology' (i.e., 'technology_systems') that is 0.75. In general, we hypothesize that a general term has lower MSN compared to a specific term.

As another metric to estimate the specificity of a term based on its $\varepsilon - neighborhood$, we define *Neighborhood Vector Similarity (NVS)*. To calculate this metric for a given term t, we first aggregate the terms in the neighborhood of t by adding their embedding vectors and then measure the semantic relatedness of the aggregated vector (denoted as $V_N(t)$) and t. Calculating the semantic relatedness between $V_N(t)$ and t could be beneficial since we consider both geometric properties of neighbors while generating $V_N(t)$ and inter-similarity associations of neighbors.

Similar to Neighborhood Vector Similarity (NVS), we define the *Neighborhood Vector Magnitude (NVM)* specificity metric. We first generate $V_N(t)$ by summing up all the neighbors' vectors in $\varepsilon - neighborhood$. Further, we measure the magnitude of $V_N(t)$. The idea behind this metric is based on two considerations: (1) in the local neighborhood of a general term such as t_j, the neighbors are less similar to each other compared to neighbors of t_i, and (2) the semantic relatedness of two vectors could be a reason for them to be put in close vicinity.

3.2.2. Graph-based

Given an $ego - network$ of a term t, while neighborhood-based metrics only focus on the connections between t and its neighborhood terms, in graph-based metrics, we take all the connections in the $ego - network$ into account. Our intuition for defining these metrics is that not only a specific term is surrounded by a higher number of neighbors in the embedding space, but also its neighbors are also highly similar to each other. We formalize these intuitions by defining $ego - network$ based on the metrics summarized in Table 2. These metrics are divided into two categories (1) *Node influence metrics* that measures the specificity of a term based on its node influence in its $ego - network$ and (2) *Edge influence metrics* that focus on the edge weights in the $ego - network$ of a term to measure its specificity.

Node influence metrics Centrality is a measure of the influence of a node in a graph. Therefore, similar to Benz et al. (2011), we consider the three most common centrality measures, including *Betweenness Centrality (BC), Degree Centrality (DC)*, and *Closeness Centrality (CC)* to estimate the influence of the ego term in the $ego - network$ and consequently estimate its specificity (Segarra & Ribeiro, 2016). As another metric, we consider *PageRank (PR)* that measures the influence of a node in a graph recursively. Further, by generalizing the idea of Inverse Document Frequency (IDF), which is a frequency-based specificity metric, we propose *Inverse Edge Frequency (IEF)* as another specificity metric as follows:

$$IEF(t) = log(\frac{|\mathbb{E}_{ego}|}{|\mathbb{E}_{ego(t)}|}) \tag{2}$$

where $\mathbb{E}_{ego}$ is the set of all edges in the $ego - network$ and $\mathbb{E}_{ego}(t)$ is set of all edges in the graph that includes the term t.

Edge influence metrics Based on the idea that the $ego - networks'$ density can be an indicator of specificity, we define *Edge Count*

Table 2
The set of proposed graph-based specificity metrics.

	Metric name	Description
Node Influence	Betweenness Centrality (BC)	Betweenness centrality of node t in the ego network
	Degree Centrality (DC)	Degree centrality of node t in the ego network
	Closeness centrality (CC)	Closeness centrality of node t in the ego network
	Inverse Edge Frequency (IEF)	The Number of edges in the ego network divided by the number of edges connected to node t in the ego network
	PageRank (PR)	The value of PageRank of term t in the ego network
Edge Influence	Edge Count (EC)	The number of edges in the ego network
	Edge Weight Sum (EWS)	The sum of edge weights in the ego network
	Edge Weight Avg_ego (EWAe)	The average of all edge weights in the ego network
	Edge Weight Max_ego (EWMe)	The minimum edge weight in the ego network

(EC) that measures the number of edges in the graph as a specificity value for the ego node.

To incorporate edge weights in the *ego − network* to calculate its density, we consider sum, average and maximum of the edge weights and define *Edge Weight Sum (EWS), Edge Weight Average_ego (EWAe)* and *Edge Weight Max_ego (EWXe)*, respectively.

3.2.3. Cluster-based metrics

These metrics are based on the idea that characteristics of term clusters within the neighborhood of a given term are potential indicators of its specificity. To calculate these metrics for a given term *t*, we first extract the term clusters around *t* by applying a clustering algorithm on the embedding vectors of terms in its $\varepsilon - neighborhood$, i.e., $N_\varepsilon(t)$, which results in *K* term clusters for *t*, i.e., $C_t^1, ..., C_t^K$. Then, we estimate the specificity of the term *t* by calculating different statistical measures on the extracted term clusters.

As mentioned before, a generic term is more likely to be related to many terms from different topics. If we cluster the neighborhood terms of a given term, we may expect that each cluster will be associated with one topic.

The association between the term clusters can also be considered to be an indicator of specificity. Each cluster is defined with its *centroid c*, which is a vector in the embedding space that indicates the center of the cluster. Therefore, for term *t*, given its term clusters, i.e., $C_t^1, ..., C_t^K$, we define its *Centroid Network*, denoted $\xi(t)$, as follows:

Definition 2. (Centroid Network) A centroid network for term *t*, denoted as $\zeta(t) = (\mathbb{V}_{cnt}, \mathbb{E}_{cnt}, g)$, is a weighted undirected graph in which $\mathbb{V}_{cbt}$ includes the centroid points of the term's clusters $C_t^1, ..., C_t^K$, and $\mathbb{E}_{cnt} = \{e_{c_i,c_j}: \forall\ c_i, c_j \in \mathbb{V}_{cnt}\}$. The weight function $g: \mathbb{E}_{cnt} \rightarrow [0, 1]$ is the cosine semantic relatedness between the vectors of two centroid points of an edge e_{c_i,c_j}, i.e., v_{c_i} and v_{c_j}.

The idea is that the more specific the terms are (1) the more the term clusters of a given term are similar, and (2) there exists a relatively dominant cluster among the term clusters (Roy et al., 2019). Therefore, edge weights in the centroid network play a crucial role in estimating specificity because they show how clusters are distributed in the embedding space. Thus, we define three metrics *Edge Weight Average_cluster (EWAc), Edge Weight Min_cluster (EWNc)* and *Edge Weight Max_cluster (EWXc)* by aggregating edge weights in the centroid network using average, min and max functions, respectively.

To exemplify the comparison of the cluster-based metrics for a generic term, e.g., 'Technology' and a more specific one, such as 'iPhone', we illustrate their clusters and centroid networks in Fig. 3. It is shown that clusters of a specific term are positioned closer to one another in the embedding space, which implies higher semantic relatedness. This is confirmed by the Edge Weight Average cluster metric, which is 0.73 for the centroid network of the term 'Technology' and 0.87 for the term 'iPhone'.

According to Fig. 3 and the intuition behind it, we define the *Cluster Elements Variance (CEV)* metric by the number of members in each cluster which has the potential to be an indicator for specificity. More specifically, a specific term has one main aspect to focus on, while a generic term, might be surrounded by multiple aspects. We elaborate on this idea by using the following example: given that we have a constant number of clusters, if a term is specific, it should have one dominant cluster in its neighborhood, e.g., the network in Fig. 3(b). Here, the blue cluster has a greater number of members compared to the other clusters in 3(b). Whereas, if a term is a generic one, it can have multiple meanings; therefore, the number of elements in the clusters should be more balanced such as shown in Fig. 3(a). We can measure the distribution of members in the clusters using variance. Thus, low variance among several members in the clusters shows that the word is an ambiguous one while high variance represents the unbalanced distribution of

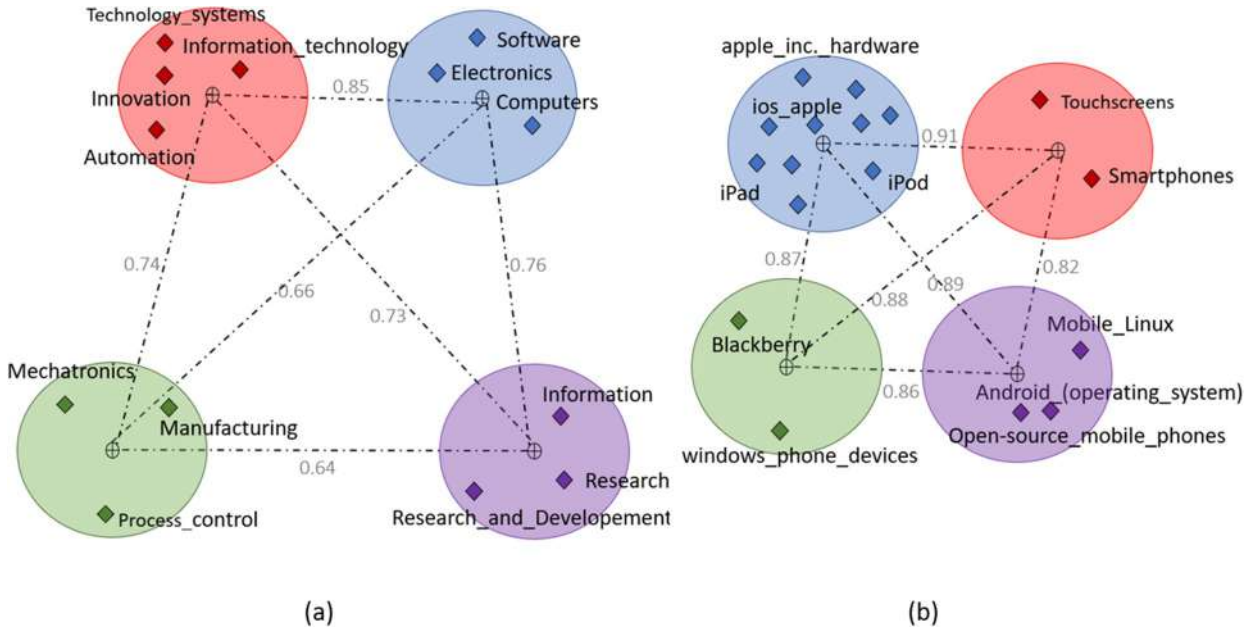

Fig. 3. The term clusters and the centroid network for (a) the generic term 'Technology' and (b) the specific term 'iPhone'.⊕ represents the centroid of each cluster. The edges in the centroid network are based on the distance between the two centroids.

Table 3
The set of proposed cluster-based specificity metrics.

Metric name	Description
EdgeWeight Avg_centroid (EWAc)	The average edge weight in the centroid network of the term t
Edge Weight Min_centroid (EWNc)	The minimum edge weight in the centroid network of the term t
Edge Weight Max_centroid (EWMc)	The maximum edge weight in the centroid network of the term t
Clusters Elements Variance (CEV)	The variance of the number of elements in the extracted clusters of the $ego-network$ of term t
Non-Specific Clustering (NSC)	Number of clusters that can be identified in the $ego-network$ of term t

members in the clusters indicating the presence of a dominant cluster and therefore the term is more specific.

So far, we have considered clustering in the neighborhood of a term in an embedding space with a defined number of clusters. In the *Non-specific Clustering (NSC)* metric, unlike EWAc, EWNc, EWMc, and CEV, we apply clustering without a pre-defined number of clusters. We use hierarchical clustering to cluster the neighbors, and then the number of the generated clusters can be used as a specificity metric. A reason for this hypothesis is that a higher number of clusters shows different aspects of the term t, i.e., term t is a more general concept and does not exclusively focus on one aspect. The set of our proposed cluster-based specificity metrics are shown in Table 3.

4. Experiments

In this section, we first evaluate the proposed specificity metrics based on test collections that consists of specificity values that can be used for directly measuring accuracy of the specificity metrics. Then, we compare our specificity-based pre-retrieval QPP methods with the state-of-the-art by predicting TREC topics performance.

4.1. Evaluation of specificity metrics

Adopted from Benz et al. (2011) and Orlandi et al. (2013), to evaluate the specificity metrics, we employed knowledge hierarchies as the gold standard. This is based on the idea that the level of each node in a knowledge hierarchy is an appropriate indicator of the node's specificity with regards to its parent and child nodes. In the following, we first describe our test collection that we used to build a gold standard and then we do a comparative analysis to evaluate the proposed specificity metrics. Finally, we train a model to incorporate all the proposed specificity metrics and explore whether the proposed metrics can serve as features in a supervised framework.

4.1.1. Test collection and experimental setup

Given a knowledge hierarchy consisting of different terms, the length of the shortest path from a term to the root of the hierarchy can be considered as the specificity of that term. Therefore, in order to avoid the biases associated with manually curated gold standard datasets for the purpose of assessing the specificity metrics, we introduced two different test collections based on the Wikipedia category hierarchy and DMOZ taxonomy.[1] The most generic term of the hierarchy sits at the topmost level, and the most specific terms are located at the leaves of the hierarchy. Adopted from Arabzadeh et al. (2019), we employed the randomly sampled unique paths, each with a length of 5, which form our test collection. Given each path in the test collection that includes a set of terms, the objective of an effective specificity metric would be to produce the correct specificity-based ordering of these terms, where the correct ordering is the one defined by the path (i.e. the hierarchy that the path originates from). It is then possible to evaluate the performance of a specificity metric by calculating the rank correlation between the actual order of terms in the path and the ranked list produced by the specificity metric. Kendall's Tau is the evaluation metric used in our experiments for measuring the rank correlation between two ranked lists. We note that in our reported results, statistical significance was measured based on a paired t-test with 95% confidence level.

In the following, we explain in more details about how we have constructed our test collection using Wikipedia category hierarchy and DMOZ taxonomy.

Wikipedia-based test collection We used the freely available English version of Wikipedia dumps[2] which consists of 1,411,022 categories with 2,830,740 subcategory relations between them. Then, to transform Wikipedia category structure into a strict hierarchy and remove its cyclic references between categories, we adopted the approach applied in Kapanipathi, Jain, Venkatramani, and Sheth (2014) and Zarrinkalam, Fani, Bagheri, and Kahani (2017). After removing the Wikipedia admin categories, we selected 'Category: Main Topic Classifications', as the root node of the hierarchy and assigned the hierarchical level to each category based on the length of its shortest path to the root node. As the last step, we removed all directed edges from a category of lower hierarchical level (specific) to categories at higher hierarchical levels. The outcome of this process is a hierarchy with a height of 26 and 1,016,584 categories with 1,486,019 links among them.

DMOZ-based test collection DMOZ, also known as the Open Directory Project (ODP), is the most comprehensive human-edited

[1] https://dmoz-odp.org/.
[2] http://wiki.dbpedia.org/Downloads2015-10.

directory of the Web. DMOZ is organized as a tree-structured taxonomy with 16 top categories and over 1,014,849 categories in-depth. We utilized the DMOZ taxonomy to assess the specificity of categories by looking at their position in the DMOZ hierarchical structure. In order to generate paths from DMOZ, we removed less meaningful categories such as those that are categorized by alphabet.

Setup In order to estimate the specificity of terms, the proposed metrics rely on the terms' neural embedding vectors. In our experiments, we utilized pre-trained neural embedding models to extract the embedding vector of each category in the test collections. Li et al. (2016) have proposed a method that simultaneously learns entity and category embedding vectors from Wikipedia. Therefore, for Wikipedia-based test collection, we adopt their trained Hierarchical Category Embedding (HCE) model. Further, in order to investigate the robustness of our proposed specificity metrics on different embedding methods, we have conducted experiments on *fastText* pre-trained embeddings on Wikipedia content (Bojanowski, Grave, Joulin, & Mikolov, 2017; Mikolov, Grave, Bojanowski, Puhrsch, & Joulin, 2018). Thus, based on the pre-trained neural embedding model utilized to extract the required embedding vectors of categories, we have two variants, namely *Wikipedia-based (HCE)* and *Wikipedia-based (fastText)*.

For *DMOZ-based* test collection, to calculate the specificity of each category, we utilized the pre-trained embedding trained on Google News[3] (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013b). The reason for selecting Google News is that DMOZ is composed of general web categories and thus required an embedding model which is not biased to any specific subject.

After removing the categories that lacked the corresponding embedding vectors in the pre-trained neural embedding models, we ended up with 713 and 170 unique paths with length of 5 in our Wikipedia-based and DMOZ-based test collections, respectively. Our constructed test collections are made publicly available.[4]

4.1.2. Comparative analysis

We evaluate all the proposed neighborhood-based, graph-based, and cluster-based specificity metrics based on three different test collections i.e, Wikipedia-based (HCE), Wikipedia-based (fastText) and DMOZ-based. The results based on the Kendall Tau rank correlation are reported in Tables 4, 5 and 6 respectively. In each table, the top-3 metrics for each test collection are highlighted. It should be noted that to extract the $\varepsilon - neighborhood$ of each term and build the $ego - network$ in the embedding space, the value of ε is set based on five-fold cross-validation optimized for Kendall Tau. Further, we used K-means as a clustering algorithm to identify clusters in $ego - network$ for calculating our cluster-based metrics ($k = 5$, we note that other values for k did not impact our findings).

Based on the results reported in Table 4, Neighborhood-based metrics perform modestly on the three test collections and among them, Weighted Degree Centrality (WDC) works the best. This confirms the hypothesis that higher WDC can be interpreted as higher specificity. Based on the results reported in Table 5, there is a discrepancy among tge metrics in the Graph-based category. Some metrics have satisfactory performance in this category whereas others have lower performance. It is observed that CC (constructed based on the centrality idea) and IEF (constructed based on the IDF idea), as two *node influence metrics*, and EWAe, as an *edge influence metric*, are among the top-3 graph-based metrics. Based on the results reported in Table 6, among all the cluster-based metrics, EWAc, EWNc, and EWXc, which are a different variation of the same idea on the centroid network, perform well and the rest of them do not show acceptable performance.

The top-3 specificity metrics based on each test collection are reported in Table 7. Based on the results, most of the selected metrics belong to the graph-based metrics. This means that graph-based metrics, which consider the relation between the neighbors of a term in addition to the relation between the term and its neighbors, lead to a more accurate estimation of the specificity.

4.1.3. Learning specificity

All the metrics introduced in previous sections estimate the specificity of a term in an unsupervised manner. In this section, we explore whether the proposed unsupervised metrics can serve as features in a supervised framework by applying a learning-to-rank strategy to collectively incorporate all the unsupervised specificity metrics to predict the specificity of a term. Due to the advantage of learning-to-rank strategy to combine a large number of features, it has attracted attention for different applications (Oosterhuis & de Rijke, 2018; Zhang & Balog, 2018). We also take advantage of the learning to rank strategy to predict the specificity of terms by representing each term by a vector of features. In our model, features are the same as the specificity metrics introduced in Section 3.2.

To apply learning to rank algorithm, we used RankLib,[5] which includes the implementation of different learning to rank methods. In this section, we first exploit two well-known and frequently used learning-to-rank methods, including RankBoost and RandomForest to find the best method for ranking the categories based on their specificity by optimizing nDCG@5. Table 8 reports the performance of the models obtained by measuring the Kendall Tau rank correlation between the actual order of categories observed in the three test collections, i.e. Wikipedia-based (HCE), Wikipedia-based (fastText) and DMOZ-based, and the list of categories ranked based on the output of each learning to rank model.

It should be noted that the results reported in Table 8 are reported by applying five-fold cross-validation for each learning-to-rank method. For easier comparison with the results of running each metric separately as an unsupervised method, Table 8 also includes the results of the Top-5 specificity metrics (based on the results reported in Section 4.1.2). As Table 8 shows, applying learning-to-rank methods (i.e., RankBoost and Random Forest) that incorporate all of the proposed specificity metrics in a single model can lead to more accurate specificity metrics compared to running each metric separately.

[3] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing.
[4] https://github.com/WikipediaHierarchyPaths/WikiPedia_Hierarchy_Paths.
[5] https://sourceforge.net/p/lemur/wiki/RankLib/.

Table 4
The performance of the neighborhood-based specificity metrics in terms of Kendall Tau rank correlation. All values are statistically significant at $\alpha = 0.05$ except the italic one.

	Neighborhood-based Metrics						
Test Collection	NS	WDC	MAD	NV	MSN	NVS	NVM
Wikipedia-based (HCE)	0.146	0.322	0.179	0.145	0.142	0.195	0.177
Wikipedia-based (fastText)	0.086	**0.249**	**0.176**	**0.176**	0.088	0.175	0.168
DMOZ-based	0.06	**0.309**	0.189	0.193	*0.016*	**0.216**	**0.291**
Average	0.097	**0.293**	0.181	0.171	0.082	**0.195**	**0.212**

Table 5
The performance of the graph-based specificity metrics in terms of the Kendall Tau rank correlation. All values are statistically significant at $\alpha = 0.05$.

	Graph-based Metrics								
	Node-influence					Edge-influence			
Test Collection	BC	CC	DC	IEF	PR	EWS	EC	EWAe	EWXe
Wikipedia-based (HCE)	0.137	**0.339**	**0.341**	**0.336**	0.131	0.139	0.207	0.315	0.291
Wikipedia-based (fastText)	0.093	**0.247**	0.217	**0.257**	0.164	0.182	0.234	**0.246**	0.174
DMOZ-based	0.250	**0.339**	0.320	**0.328**	0.279	0.162	0.220	**0.379**	0.318
Average	0.160	**0.292**	**0.292**	**0.307**	0.191	0.161	0.220	**0.313**	0.261

Table 6
The performance of the cluster-based specificity metrics in terms of the Kendall Tau rank correlation. All values are statistically significant at $\alpha = 0.05$.

	Cluster-based Metrics				
Test Collection	CEV	EWAc	EWNc	EWXc	NSC
Wikipedia-based (HCE)	0.213	**0.320**	**0.321**	**0.295**	0.116
Wikipedia-based (fastText)	0.087	**0.186**	**0.162**	**0.170**	0.129
DMOZ-based	0.213	**0.409**	**0.419**	**0.371**	0.369
Average	0.171	**0.305**	**0.300**	**0.278**	0.204

Table 7
Top-3 best-performing metrics based on each test collection.

Wikipedia-based (HCE)	Neighbourhood-based –	Graph-based CC, DC, IEF	Cluster-based –
Wikipedia-based (fastText)	WDC	IEF, CC	–
DMOZ-based	–	EWAe	EWAc, EWNc

Table 8
The performance of learning-to-rank models and the top-5 best-performing metrics (based on their average performance on all the test collections reported in Section 4.1.2), in terms of Kendall Tau rank correlation. All the results are statistically significant at $\alpha = 0.5$.

	Supervised methods		Unsupervised methods				
	RankBoost	Random Forest	WDC	EWAe	IEF	EWAc	EWNc
Wikipedia-based (HCE)	0.350	**0.409**	0.322	0.315	0.336	0.320	0.321
Wikipedia-based (fastText)	0.389	**0.426**	0.249	0.246	0.257	0.186	0.162
DMOZ-based	0.429	**0.521**	0.309	0.379	0.328	0.409	0.419
Average	0.389	**0.452**	0.293	0.313	0.307	0.305	0.300

4.1.4. Comparison with baselines

We compare the performance of our specificity metrics to the performance of two well-known frequency-based specificity metrics, i.e., Average IDF and Simplified Clarity Score (SCS), as the baselines. In order to apply the frequency-based specificity metrics, i.e., SCS and IDF, to estimate the specificity of Wikipedia categories, we consider Wikipedia as a collection of documents (each Wikipedia

article is a document). Further, to apply the frequency-based metrics on DMOZ categories, we utilized the dataset[6] which was retrieved with a crawler in 2006 from the Open Directory Project (ODP) by selecting categories from the ODP hierarchy while imposing some constraints to ensure the quality of the dataset which is restricted to English. For each topic, we collected all of its URLs as well as those in its subtopics. The total number of collected pages, after retrieving, parsing and cleaning the HTML was more than 350K from 448 topics.

As another baseline metric, we adopt the recently proposed approach by Roy et al. (2019) that utilizes neural embedding-based word vectors to predict query performance. Their specificity metric, known as $P_{clarity}$, is based on the idea that the number of clusters around the neighborhood of a term t is a potential indicator of its specificity. To apply their specificity metric, we have used the implementation provided by the authors.

In Table 9, the results of the baselines are compared to the best variations of our supervised and unsupervised models. By comparing the overall performance of our proposed cluster-based method and $P_{Clarity}$, it can be observed that $P_{Clarity}$ performs better than all of our cluster-based methods. This suggests that the characteristics of the dominant term cluster around a term is a better clue for estimating term specificity. However, in this work we proposed other specificity metrics, based on a network representation of neural embeddings,such as our graph-based metrics that still performed better than $P_{Clarity}$. As shown, both of our supervised and unsupervised methods outperform the baselines despite the fact that the baseline methods have access to corpus-specific frequency information whereas our methods do not and are solely based on pre-trained neural embeddings.

4.1.5. Data analysis

In this section, we conduct additional in-depth analysis to empirically show the performance of our proposed specificity metrics. Overall, our specificity metrics, described in Section 3, are based on four main assumptions, as reported in Table 10. In this table, the formalization of each assumption is also provided.

In order to examine these assumptions, we applied the following steps on our Wikipedia test collection, which includes 713 paths with a length of 5, each of which is ordered from the most generic to the most specific. First, for each path, we selected the first and last term as the most generic and the most specific term, respectively. This process resulted in two groups of terms: (1) group S composed of 713 specific terms, and (2) group G containing 713 generic terms. For instance, given path P which includes the following terms: 'Food and Drink', 'Fast food', 'Fast-food Restaurant', 'Fast-food chains in the US" and 'Pizza Hut', the first term in the path, i.e. 'Food and Drink', is selected as the most generic term for group G and the last term, i.e. 'Pizza Hut' is selected as the most specific term in the path for group S.

In order to confirm each assumption, we studied the assumption on each group of terms separately and displayed the distribution of values in each group using boxplot graphs. Fig. 4 depicts the results for the four assumptions. For example, regarding Assumption 1, for each group of terms, we first measured the number of neighbors in $\varepsilon - neighbourhood$ of each term in that group and then visualized the results using a boxplot for each group separately. A boxplot is a standardized way of displaying the distribution of data based on minimum, first quartile, median, third quartile and the maximum point of the data. Therefore, we can easily investigate each assumption by comparing the box plot of group S and group G.

Based on Fig. 4, for all the four assumptions, we can observe that the range of values (i.e., first quartile, median, third quartile) for group S is different from group G, which confirms our assumptions. For instance, Fig. 4 shows that the number of terms in $\varepsilon - neighbourhood$ of terms in group S is higher than the number of terms in $\varepsilon - neighbourhood$ of group G, which supports Assumption 1. Similarly, in confirmation of Assumption 2, we can conclude from Fig. 4 that the more the average similarity of terms in $\varepsilon - neighbourhood$ to term t is, the more specific the term t is. This fact is also true for average pair-wise similarity of all terms in $\varepsilon - neighbourhood$ (except the term of interest), i.e., a more specific term has a higher average pair-wise similarity of all terms in its $\varepsilon - neighbourhood$ compared to a more generic term. In addition, by comparing the boxplot for Assumption 4 for groups S and G, we can conclude that the higher the variance of the similarities of terms in $\varepsilon - neighbourhood$ of term t is, the more generic term t would be.

4.2. Evaluation of pre-retrieval QPP

In this section, we aim at evaluating the proposed specificity-based pre-retrieval QPP methods by predicting TREC topics' performance. A common approach for measuring the performance of a QPP method is to use correlation metrics to measure the correlation between the list of queries ordered by their difficulty for the retrieval method and the list of queries ordered by the QPP method. As two most common correlation metrics in this area, we used Kendall's τ and Pearson's ρ coefficient, which measures ranking and linear correlation, respectively.

4.2.1. Test collection and experimental setup

We employed three widely used corpora for evaluating pre-retrieval QPP methods, namely Robust04, ClueWeb09, and GOV2. For Robust04, TREC topics 301–450 and 601–650, for ClueWeb09, topics 1–200, and for GOV2, topics 701–850 are used. The topic difficulty was based on Average Precision (AP) of each topic computed using Query Likelihood (QL) implemented in Anserini[7] for the three collections. Robust04 consists of 528,155 documents, ClueWeb09 consists of 50 million English web pages and Gov2 consists of

[6] https://data.mendeley.com/datasets/9mpgz8z257/1.
[7] https://github.com/castorini/anserini.

Table 9
Performance of learning-to-rank models in terms of Kendall Tau rank correlation and our proposed best-performing metrics. All the results are statistically significant at $\alpha = 0.5$.

	Baselines			Our Method (best variations)	
	avgIDF	SCS	$P_{Clarity}$	Unsupervised (EWAe)	Supervised (Random Forest)
Wikipedia-based (HCE)	0.304	0.144	0.121	0.315	**0.409**
Wikipedia-based (fastText)				0.246	**0.426**
DMOZ-based	0.369	0.348	0.179	0.379	**0.521**
Average	0.336	0.246	0.150	0.339	**0.452**

Table 10
Four main assumptions of our proposed Specificity, *Spec*(*t*), metrics.

#	Description	Formula
1	The number of neighbors in ε − *neighborhood* of term t is an indicator of the specificity of term t.	$Spec(t) \propto \lvert N_\varepsilon(t)\rvert$
2	The average similarity of term t to all terms in ε − *neighborhood* is an indicator of the specificity of term t.	$Spec(t) \propto \frac{\sum_{n_i \in N_\varepsilon(t)} Sim(t, n_i)}{\lvert N_\varepsilon(t)\rvert}$
3	The average pair similarity of all terms in ε − *neighborhood* (except the term of interest) is an indicator of the specificity of term t.	$Spec(t) \propto \frac{2(\sum_{n_i, n_j \in N_\varepsilon(t))} Sim(n_i, n_j)}{\lvert N_\varepsilon(t)\rvert (\lvert N_\varepsilon(t)\rvert - 1)}$
4	The variance of similarity of term t to all terms in ε − *neighborhood* is an indicator of the specificity of term t.	$Spec(t) \propto Var(\{Sim(t, n_i) \lvert n_i \in N_\varepsilon(t)\})$

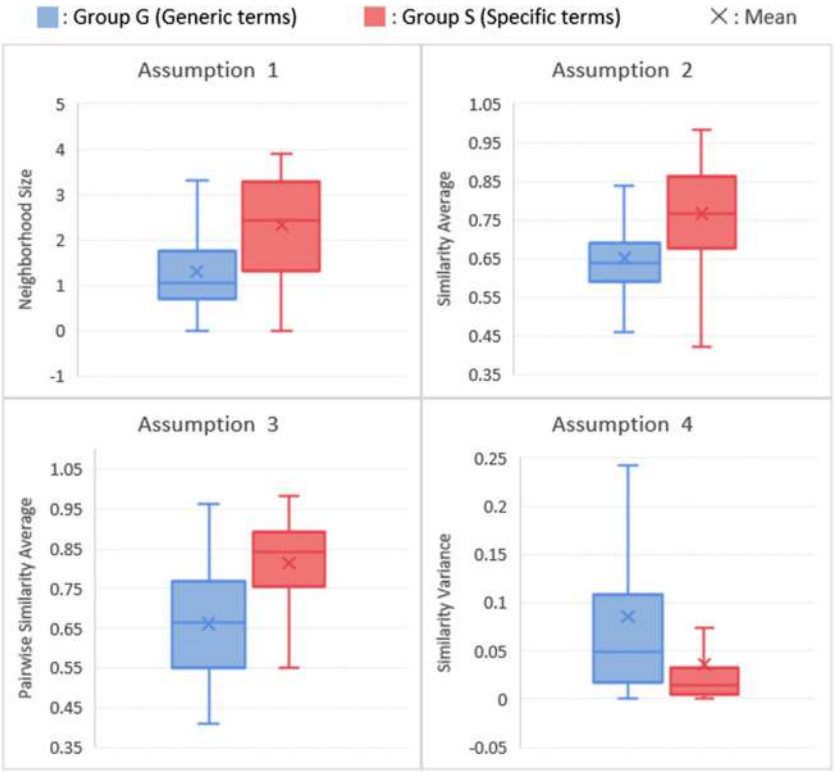

Fig. 4. Distribution of the normalized values in groups G and S using a boxplot.

over 25 million documents.

In our proposed pre-retrieval QPP approach, to estimate the specificity of the query terms, we have utilized the embedding model pre-trained on Google News (Mikolov et al., 2013b). Further, our metrics are dependent on some hyper-parameters, which were set using five-fold cross-validation optimized for Pearson correlation.

It should be noted that since some of the specificity metrics are defined for a single term, to estimate the specificity of a query composed of more than one term, we used aggregation functions (i.e., sum, avg, min, and max) (Hauff, Kelly, & Azzopardi, 2010) over the specificity value of its constituent terms.

Table 11
Frequency-based pre-retrieval QPP baselines.

Baseline	Formula	Description
SCQ	$SCQ(t) = (1 + \log(TF(t, D))).\ IDF(t)$	D denotes the collection. $TF(t, D)$ denotes the term frequency of term t in collection D.
IDF	$IDF(t) = \log(\frac{N}{N_t})$	N denotes the number of documents in the collection. N_t is the number of documents containing query term t.
SCS	$SCS(q) = \log(\frac{1}{\|q\|}) + avgICTF(q)$	$avgICTF(q) = \frac{1}{\|q\|}\sum_{t\in q}\log(\frac{\|D\|}{TF(t.D)})$
PMI	$PMI(t_i, t_j) = \log\frac{P_r(t_i, tj \mid D)}{P_r(t_i \mid D)P_r(t_j \mid D)}$	$P_r(t_i, t_j\|D)$ denotes the probability of two terms co-occurring in the collection.
VAR	$VAR(w(t, d))$	$VAR(w(t, d))$ is the variance of term weights over documents $d \in D$ containing query term t, where : $w(t, d) = \frac{\log(1 + TF(t,d)).IDF(t)}{\|d\|}$

4.2.2. Baselines

For comparative analytics, we adopt Collection Query Similarity (SCQ) (Zhao et al., 2008), Inverse Document Frequency (IDF) (He et al., 2008), Simplified Clarity Score (SCS) (He et al., 2008), Point-wise Mutual Information (PMI) (Hauff, 2010) and term weight Variance (VAR) (Zhao et al., 2008), which are the most widely used metrics reported in the QPP literature (Carmel & Yom-Tov, 2010). The metrics are formalized in Table 11.

As another baseline, we considered the recent work proposed by Roy et al. (2019). They have proposed a novel query performance predictor, which utilizes neural embedding representation of terms. Their work is based on the assumption that higher probability of possible clusters around a term indicates the more generality of the term. Therefore, they utilized a Gaussian Mixture Model (GMM) in order to find the probability of different clusters around a term in a local neighbourhood of terms within embedding space. Given that each Gaussian component potentially corresponds to a different sense of a term, they calculate $P_{clarity}$ using prior probability of choosing the most dominating sense of the query term and the posterior probability of sampling the term embedding vector from the selected component of the GMM. We applied their method using embedding model pre-trained on Google News (Mikolov et al., 2013a).

Table 12
The average Pearson and Kendall Tau correlation results of QPP on TREC TOPICS 301–450 and 600–650 on Robust 04, 1–200 on Clueweb and 700–850 on GOV2.

		Method	Robust04		ClueWeb09		GOV2		All corpora	
			Pearson	Kendall	Pearson	Kendall	Pearson	Kendall	Pearson	Kendall
Baselines		maxVAR	0.33	**0.31**	0.17	0.20	0.14	0.09	0.21	0.20
		maxSCQ	0.38	**0.33**	0.22	0.19	**0.40**	**0.29**	0.33	**0.27**
		avgIDF	**0.43**	0.28	0.17	0.18	0.27	0.21	0.29	0.22
		SCS	0.39	0.25	0.15	0.16	0.22	0.18	0.25	0.19
		maxPMI	0.14	0.15	0.10	0.10	0.31	0.22	0.18	0.16
		P_clarity	0.32	0.23	0.25	0.20	0.35	0.25	0.30	0.22
Neighbourhood-based		avgNS	0.22	0.14	0.10	0.09	0.09	0.08	0.14	0.10
		sumWDC	0.25	0.10	0.12	0.07	0.12	0.11	0.17	0.09
		maxMAD	0.23	0.14	0.11	0.07	0.12	0.06	0.15	0.09
		sumNV	0.28	0.21	0.13	0.11	0.14	0.07	0.18	0.13
		sumMSN	0.28	0.13	0.11	0.10	0.15	0.12	0.18	0.12
		sumNVS	0.25	0.18	0.11	0.07	0.12	0.07	0.16	0.10
		maxNVM	0.21	0.13	0.11	0.07	0.11	0.09	0.14	0.09
Graph-based	Node-influence	maxBC	0.33	0.31	0.31	**0.25**	0.32	0.20	0.32	0.25
		avgCC	0.38	0.29	**0.32**	**0.27**	**0.36**	**0.29**	**0.35**	**0.28**
		maxDC	**0.40**	0.28	**0.32**	**0.26**	**0.36**	**0.30**	**0.36**	**0.28**
		avgIEF	**0.42**	**0.37**	0.27	0.23	0.33	0.23	**0.34**	**0.27**
		maxPR	0.36	0.26	**0.34**	0.21	0.34	0.21	**0.34**	0.23
	Edge-Influence	minEWS	0.39	0.23	0.19	0.16	0.30	0.24	0.29	0.21
		minEC	0.36	0.20	0.19	0.16	0.24	0.23	0.26	0.19
		avgEWAe	0.32	0.21	0.13	0.09	0.14	0.14	0.20	0.14
		avgEWXe	0.29	0.16	0.13	0.07	0.13	0.12	0.18	0.12
Cluster-based		CEV	0.25	0.10	0.09	0.06	0.11	**0.31**	0.15	0.16
		EWAc	0.34	0.20	0.14	0.07	0.19	0.12	0.22	0.13
		EWNc	**0.39**	0.25	0.15	0.10	0.19	0.13	0.24	0.16
		EWXc	0.31	0.18	0.11	0.10	0.16	0.10	0.19	0.13
		NSC	0.30	0.17	0.17	0.11	0.14	0.08	0.20	0.12

Table 13
Pearson Correlation results of QPP on Robust04, ClueWeb09 and GOV2. *Indicates statistical significance at $\alpha = 0.05$.

		Method	Robust04				ClueWeb09				GOV2		
			300-350	351-400	401-450	600-650	1-50	51-100	101-150	151-200	701-750	751-800	801-850
Baselines		maxVAR	0.08	**0.46***	0.23	**0.55***	0.14	0.04*	**0.42***	0.08	0.27*	0.06*	0.1
		maxSCQ	0.01	0.5*	**0.66***	0.35*	0.22	0.24*	0.33*	0.09	**0.53***	0.32*	0.35*
		avgIDF	**0.52***	0.42*	**0.43***	0.36*	0.18	0.21*	0.27	0.01	0.4*	0.25*	0.17*
		SCS	0.43*	0.35*	0.34*	0.42*	0.18	0.19*	0.16	0.05	0.34*	0.19*	0.12
		maxPMI	0.06	0.12	0.28*	0.08	0.2*	0.12	0.04	0.04	**0.44***	0.26*	0.22
		$P_{clarity}$	0.31*	0.24*	0.36*	0.38*	**0.29***	0.27*	0.18	0.24	0.33*	0.33*	0.38*
Neighbourhood-based		avgNS	0.02	0.4*	0.05	0.39*	0.08	0.13	0.11	0.08	0.11	0.08	0.09
		sumWDC	0.47*	0.01	0.40*	0.13	0.18	0.13	0.15	0.03	0.11	0.01	0.24
		maxMAD	0.09	0.19	0.35*	0.28*	0.08	0.08	0.14	0.13	0.21	0.01	0.15
		sumNV	0.40*	0.24	0.18	0.31*	0.04	0.13	0.27	0.06	0.01	0.12	0.28*
		sumMSN	0.48*	0.04	0.42*	0.18	0.18	0.11	0.13	0.01	0.16	0.04	0.26*
		sumNVS	0.42*	0.06	0.36*	0.14	0.12	0.17	0.11	0.04	0.08	0.04	0.24*
		maxNVM	0.10*	0.31*	0.39*	0.04	0.06	0.01	0.22	0.13	0.17	0.13	0.02
Graph-based	Node-influence	maxBC	0.35*	0.17	0.39*	0.40*	0.28*	**0.29***	0.37*	0.28*	0.18	0.28*	**0.5**
		avgCC	0.24*	0.41*	0.42*	**0.45***	**0.31***	0.17	**0.47***	**0.33***	0.14	**0.42***	**0.51***
		maxDC	0.28*	0.42*	**0.44***	0.44*	0.25*	**0.31***	**0.37***	**0.34***	0.15*	**0.41***	**0.52***
		avgIEF	0.41*	0.44*	0.40*	0.43*	**0.29***	0.26*	0.17	**0.36***	0.18	**0.33***	0.47*
		maxPR	0.33*	**0.47***	0.31*	0.31*	**0.36***	**0.31***	0.37*	0.31*	**0.38***	0.32*	0.32*
	Edge-Influence	minEWS	0.34*	**0.49***	0.26*	**0.46***	0.07	0.27	0.27	0.14	0.12	**0.33***	0.44*
		minEC	0.37*	0.41*	0.26*	0.40*	0.06	0.27	0.29	0.14	0.11	0.24*	0.38*
		avgEWAe	0.33*	0.45*	0.18	0.33*	0.19	0.08	0.17	0.08	0.09	0.17	0.16
		avgEWXe	0.29*	0.30*	0.21	0.34*	0.25*	0.01	0.21*	0.03	0.06	0.15*	0.17
Cluster-based		CEV	0.15	0.38*	0.09	0.36*	0.02	0.02	0.30*	0.03	0.06	0.10	0.17
		EWAc	**0.54***	0.19	0.32*	0.29*	0.03	0.18	0.22	0.13	0.19	0.18	0.21
		EWNc	**0.54***	0.28*	0.34*	0.38*	0.18	0.12	0.24	0.06	0.16	0.23	0.18
		EWXc	**0.52***	0.15	0.29*	0.26	0.08	0.16	0.19	0.01	0.13	0.15	0.19
		NSC	0.50*	0.07	**0.43***	0.18	0.14	0.15	0.31*	0.06	0.10	0.14	0.17

4.2.3. Results

In this section, we investigate the quality of different QPP methods on different topic sets on Robust04, ClueWeb09 and Gov2 by computing Kendall τ and Pearson ρ. The average performance of each method on each corpora is reported in Table 12. More detailed results are given in Tables 13 and 14, which present Kendall tau and Pearson, respectively, for individual TREC topic sets (each set consisting of 50 queries), respectively. The top-3 best-performing methods have been highlighted in the tables.

Empirical studies on pre-retrieval QPP metrics have shown that while some metrics show better performance on some corpora and topic sets, there is no single metric or a set of metrics that outperforms the others on all topics and corpora (Carmel & Yom-Tov, 2010). The results of our experiments reported in Tables 13 and 14 confirm this as well. Therefore, to analyze the overall performance of the QPP methods, we reported the average of the results on all the corpora and topic sets in the last column of Table 12 named *All Corpora*.

Based on the results in Table 12, by comparing the overall performance of the frequency-based pre-retrieval QPP baselines (i.e., maxVAR, maxSCQ, avgIDF, SCS and maxPMI) and our embedding-based methods, we can observe that, the top-3 methods (i.e., maxDC, avgCC and avgIEF) in terms of both Pearson and Kendall Tau are among the better embedding-based methods. However, all the embedding-based methods cannot outperform all the frequency-based methods. For example maxSCQ as a frequency-based method outperforms many of the embedding-based methods (e.g., avgNS, sumMAD and maxMAD). This means that only utilizing the neural embeddings of the terms to define a QPP method does not lead to a superior method compared to the common frequency-based QPP methods and defining an inappropriate embedding-based metric may lead to a poor performance.

As another observation, by comparing three categories of the proposed specificity-based QPP methods (i.e., neighborhood-based, graph-based and cluster-based), we can observe that the graph-based methods report better results compared to most of the neighborhood-based and cluster-based methods. As a result, graph-based methods are more accurate indication to estimate the specificity of the terms. A possible explanation is that since the graph-based metrics consider all the relations in the $ego-network$ to estimate the specificity of the ego term, this makes them more accurate specificity metrics compared to neighborhood-based and cluster-based metrics, which further leads to better QPP methods. This observation was also confirmed in Section 4.1.2 based on the results reported in Table 7.

As explained in Section 3.2.2, we divided the graph-based methods into two categories *Node influence metrics* that measure the

Table 14
Kendall Tau Correlation results of QPP on Robust04, ClueWeb09 and GOV2. * Indicates statistical significance at $\alpha = 0.05$.

		Method	Robust04				ClueWeb09				GOV2		
			300-350	351-400	401-450	600-650	1-50	51-100	101-150	151-200	701-750	751-800	801-850
Baselines		maxVAR	0.17	**0.36***	**0.35***	**0.34***	**0.28***	0.23*	**0.27***	0.01	0.2*	0.02	0.05*
		maxSCQ	0.13	**0.42***	0.5*	0.26*	0.19*	0.25*	**0.30***	0.03	**0.36***	**0.29***	0.23*
		avgIDF	0.22*	0.29*	0.28*	0.32*	**0.24***	0.25*	0.16	0.05	0.27*	0.22*	0.14
		SCS	0.21*	0.24*	0.23*	0.3*	**0.23***	0.24*	0.10	0.07	0.23*	0.19*	0.11
		maxPMI	0.04	0.18	0.18*	0.2	0.15	0.16	0.07	0.03	**0.28***	0.22*	0.16*
		$P_{clarity}$	0.20*	0.20	0.25*	0.26*	0.3	0.15	0.1	0.24*	0.25*	0.24*	0.25*
Neighbourhood-based		avgNS	0.07	0.25*	0.02	0.22*	0.08	0.09	0.14	0.03	0.11	0.11	0.01
		sumWDC	0.22*	0.02	0.12	0.02	0.17	0.02	0.03	0.05	0.13	0.01	0.20*
		maxMAD	0.10	0.10	0.21*	0.16	0.02	0.07	0.07	0.12	0.11	0.03	0.05
		sumNV	0.25*	0.13	0.18	0.26*	0.05	0.16	0.21*	0.01	0.02	0.07	0.11
		sumMSN	0.24*	0.01	0.19*	0.07	0.14	0.02	0.21*	0.01	0.12	0.03	0.22*
		sumNVS	0.23*	0.03	0.18	0.26	0.05	0.14	0.04	0.03	0.06	0.08	0.06
		maxNVM	0.02	0.20*	0.08	0.20	0.07	0.03	0.12	0.04	0.12	0.09	0.06
Graph-based	Node-influence	maxBC	0.31*	0.20*	**0.30***	**0.42***	0.22*	**0.28***	0.26*	0.22*	0.18	0.11	0.3*
		avgCC	0.24*	0.29*	0.27*	**0.37***	0.22*	0.2*	**0.28***	**0.36**	0.11	**0.36***	**0.41***
		maxDC	0.20*	0.29*	**0.30***	0.33*	0.22*	**0.27***	0.15*	**0.38***	0.12	**0.36***	**0.41***
		maxPR	0.25*	**0.40***	0.24*	0.16	0.13	**0.29***	0.18	0.25*	**0.28***	0.19*	0.15
		avgIEF	**0.47***	**0.36***	0.30*	0.33*	0.17	0.2*	0.17	**0.39***	0.09	0.25*	**0.34***
	Edge-Influence	minEWS	0.17	0.24*	0.29*	0.22*	0.13	0.18	0.26*	0.06	0.2*	0.2*	0.31*
		minEC	0.17	0.15	0.27*	0.19	0.14	0.16	**0.27***	0.05	0.12	0.26*	0.3*
		avgEWAe	0.19	0.18*	0.29*	0.17	0.22*	0.03	0.03	0.07	0.10*	0.14	0.17*
		avgEWXe	0.12	0.14*	0.22*	0.15	0.15*	0.07	0.03	0.04	0.09	0.15	0.13*
Cluster-based		CEV	0.01	0.21*	0.01	0.18	0.07	0.01	0.16	0.01	0.21*	0.01	0.7
		EWAc	**0.35***	0.12	0.17*	0.17*	0.03	0.2	0.02	0.03	0.13	0.12	0.10
		EWNc	**0.37***	0.17*	0.22*	0.23*	0.09	0.0.8	0.14	0.08	0.10	0.16	0.12
		EWXc	0.32*	0.09	0.16*	0.15	0.04	0.13	0.06	0.16	0.10	0.11	0.09
		NSC	0.29*	0.04	0.25*	0.11	0.13	0.04	0.07	0.2	0.10	0.08	0.07

specificity of a term based on its node influence in its *ego − network* and *Edge influence metrics* that focus on the edge weights in the *ego − network* of a term to measure its specificity. As shown in Table 12, the top-3 methods are among the Node influence metrics. This means that node influence metrics that focus on the influence of the ego term lead to more accurate specificity metrics compared to edge-influence metrics that consider all the nodes in the *ego − network* equally and overlook the specific role of the ego term and its connected edges.

Among the baseline methods, $P_{Clarity}$ (Roy et al., 2019) is an embedding based QPP method. Similar to our cluster-based metrics, their idea is that the higher probability of possible clusters around a term indicates higher generality of the term. Whereas our cluster-based metrics consider all the term clusters around the term uniformly, they have focused only on the characteristics of the dominant cluster. By comparing the overall performance of our proposed cluster-based method and $P_{Clarity}$, it can be observed that $P_{Clarity}$ performs better than all of our cluster-based methods. This suggests that the characteristics of the dominant term cluster around a term is a better clue for estimating term specificity. However, our graph-based metrics still performed better than $P_{Clarity}$.

Overall, when considering a balance between the two evaluation measures and the performance of the metrics on all topics and corpora, we find our DC, CC and IEF metrics to be well-performing metrics across the board. It is important to note that, the added benefit of the DC metric is that it is inexpensive to compute with O(1) while CC tends to have a high time complexity of $O(V^3)$.

4.3. Performance on non-factoid questions

While QPP for ad-hoc retrieval tasks has been studied extensively in the information retrieval (IR) community, evaluating QPP methods for Question Answering (QA) systems is rather novel. The increasing use of mobile and voice search has attracted researchers' attention to the importance of QA performance prediction task (Hashemi, Zamani, & Croft, 2019b). It should be noted that QPP for ad-hoc retrieval differs from QPP for QA from various aspects such as length of answers (documents) and questions (queries). As a result, it is possible to expect different performance from QPP methods when applied to open-ended questions instead of ad-hoc retrieval queries. In this section, we evaluate our proposed neural embedding based pre-retrieval QPP methods on a non-factoid question answering collection called ANTIQUE (Hashemi, Aliannejadi, Zamani, & Croft, 2019a), which consists of 2626 questions and 34,011 answers. We compare the effectiveness of our top-3 performed pre-retrieval methods to all our baselines introduced in

Table 15
Kendall Tau and Pearson Rho Correlation results on the ANTIQUE QA collection. * Indicates statistical significance at alpha = 0.05.

	Our methods (Best Variations)			Baselines					
	avgCC	maxDC	avgIGE	maxVAR	maxSCQ	avgIDF	SCS	maxPMI	$P_{clarity}$
Pearson ρ	**0.34***	0.3*	0.31*	0.07	**0.34***	0.15*	0.2*	0.07	0.16*
Kendall τ	**0.26***	0.25*	0.21*	0.15*	0.22*	0.14*	0.1	0.06	0.14

Section 4.2.2 on the ANTIQUE QA collection. The results for the ANTIQUE test collection (200 queries) are reported in Table 15. Similar to Section 4.2, to evaluate the performance of each QPP method on the ANTIQUE QA collection, we measure the correlation between the list of questions ordered by their difficulty for the retrieval method and the list of questions ordered by the QPP method using Pearson Rho and Kendall Tau. The topic difficulty is based on Average Precision (AP) of each topic computed using BERT (Devlin, Chang, Lee, & Toutanova, 2018), which was reported as the top-performing model on the ANTIQUE collection (Hashemi et al., 2019a). Since the queries in the ANTIQUE collection are informal questions generated by real users and in order to avoid out of vocabulary problem, we used pre-trained embedding from fastText to implement the neural embedding based methods (avgCC, maxDC, avgIGE and Pclarity) reported in Table 15.

As shown in Table 15, our embedding-based methods, i.e., maxDC, avgCC and avgIEF, outperform most of the pre-retrieval QPP baselines and in terms of both Pearson Rho and Kendall Tau. Some of the frequency-based methods such as maxPMI, which consider term occurrence in documents suffer from poor performance on the QA collection due to short length of the answers. On the other hand, neural embedding-based methods that work based on the semantic aspects of the terms and ignore the frequency of terms in the corpus, are immune from the short length of answers in the QA collection.

4.4. Impact of out of vocabulary terms

In this section, we investigate the effect of out of vocabulary terms in neural embedding collections on our proposed approach. We have already shown that the proposed neural embedding-based metrics is robust to the choice of the pre-trained embeddings by assessing the metrics over three different pre-trained embeddings. Since we rely on pre-trained word embeddings to extract the embedding vectors of query terms, our approach is susceptible to the out-of-vocabulary problem, if the query term is not included in the used pre-trained embeddings. However, the degree of susceptibility of our approach depends on the pre-trained model that is used. In particular, from the persepctive of handling out-of-vocabulary terms, pre-trained word embedding models can be classified into two groups:

1. Pre-trained embedding models, like fastText, which can potentially construct a vector for unobserved terms. Therefore, by utilizing these pre-trained word embeddings our proposed pre-retrieval query performance predictors are not affected by out of vocabulary terms.
2. Pre-trained embedding models, like the one trained on Google News, which are not able to handle unobserved terms. We have conducted detailed analysis to investigate the impact of out-of-vocabulary terms on the quality of our proposed pre-retrieval query performance predictors when a pre-trained embedding model of this type is used. In Table 16, we reported the percentage of queries affected by this issue in each dataset separately, i.e., Robust04, ClueWeb09 and GOV2. The table shows that 3.8% of all the queries contained at least one term with a missing embedding vector and 0.7% of all the queries are removed because none of the terms existed in the vocabulary of the pre-trained embeddings. We conclude that by using this group of embeddings, our proposed model cannot be applied to 0.7% of queries, which affects the performance of 3.8% of queries.

To further analyze this issue, we have conducted an experiment on our top-3 metrics, i.e., CC, DC and IEF, by using both kinds of pre-trained embedding models, namely FastText and Google News embeddings. The results are reported in Tables 17 and 18 in terms of Pearson Rho and Kendall Tau on the most affected corpus, i.e., Robust04. Based on a paired *t*-test applied to the results on the two embeddings, there is no significant difference between them ($\alpha = 0.05$). Based on these results, we conclude that the performance improvements shown by our metrics are still statistically significant despite out of vocabulary terms.

4.5. Qualitative analysis

In this section, we provide further insight as to where the proposed model fails and where it is able to succeed compared to

Table 16
Statistics of query terms affected by out of vocabulary issue.

	Robust04	ClueWeb09	GOV2	All
Affected Queries	6.0% (12/200)	3.5% (7/200)	0.0% (2/150)	3.8% (21/550)
Omitted Queries	0.0% (0/200)	2.0% (4/200)	0.0% (0/150)	0.7% (4/550)

Table 17
Comparing the Pearson Rho correlation results of our top-3 pre-retrieval QPP metrics by utilizing pre-trained embedding on Google news and fastText embeddings on Robust04.

Pre-trained Embedding	Method	Robust04				
		301–350	351–400	401–450	600–650	Avg
Google News	avgCC	0.24	0.41	0.42	0.45	0.38
	maxDC	0.28	0.42	0.44	0.44	0.40
	avgIEF	0.41	0.44	0.4	0.43	0.42
fastText	avgCC	0.24	0.39	0.41	0.47	0.38
	maxDC	0.22	0.32	0.39	0.49	0.35
	avgIEF	0.38	0.35	0.36	0.47	0.39

Table 18
Comparing the Kendall Tau correlation results of our top-3 performed pre-retrieval QPP metrics by utilizing pre-trained embedding on Google news and fastText embeddings on Robust04.

Pre-trained Embedding	Method	Robust04				
		301–350	351–400	401–450	600–650	Avg
Google News	avgCC	0.24	0.29	0.27	0.37	0.29
	maxDC	0.2	0.29	0.3	0.33	0.28
	avgIEF	0.47	0.36	0.3	0.33	0.37
fastText	avgCC	0.19	0.32	0.26	0.33	0.28
	maxDC	0.18	0.29	0.26	0.36	0.27
	avgIEF	0.34	0.31	0.27	0.34	0.32

Table 19
Five sample queries from 'well-performed' group and five sample queries from 'poor-performed' group on Robust04.

Well-performed Queries	Poor-performed Queries
Risk of Aspirin	Police Deaths
British Chunnel Impact	Unsolicited Faxes
Hydrogen Energy	Illegal Technology Transfer
Dismantling Europe Arsenal	Killer Bee Attacks
Berlin Wall Disposal	Best Retirement Country

baseline methods. To achieve this objective, we first divide the TREC queries into two groups based on the performance of our top-3 methods, i.e., CC, DC, and IEF on Robust04 compared to the baselines: (1) 'Well-performed' are queries where we outperform the baselines; and, (2) 'Poor-performed' are queries where we under-perform. Five sample queries from each group are reported in Table 19.

Based on the characteristics of the queries in each group, we observed that there are more *proper nouns* among the well-performed queries compared to poor-performed ones. We believe that the reason can be that the baselines, i.e., frequency-based methods, might not be able to perform well on queries containing proper nouns since they might not occur in the corpus at all or frequently enough. On the other hand, our neural embedding-based methods are robust to corpus statistics and focus on the semantic aspects of terms.

To further analyze this observation, we conducted an experiment in which we first randomly sampled 50 queries that have *no proper nouns* and 50 queries that contain *at least one proper noun* from the TREC topics (301–450 and 601–650). Then, the performance of the best variations of our methods and baselines on these two groups of queries are reported in Table 20 in terms of Kendall Tau and Pearson correlation.

As shown in Table 20, although our proposed methods outperforms the baselines in both groups of queries, the percentage of improvement is higher in queries containing proper nouns. More specifically, by comparing the best variation of our methods, i.e., CC

Table 20
Kendall Tau and Pearson Rho Correlation results on Robust04 for 50 randomly sampled TREC queries which contain at least one proper noun and 50 randomly sampled queries containing no proper nouns. * Indicates statistical significance at alpha = 0.05.

		Our Method (Best Variations)			Baselines					
Query Group	Metrics	avgCC	maxDC	avgIEF	maxVAR	maxSCQ	avgIDF	SCS	maxPMI	$P_{clarity}$
At least one Proper noun	Pearson	**0.44***	0.35*	0.3*	0.07	0.25*	0.04	0.03	0.08	0.12*
	Kendall Tau	**0.27***	0.22*	0.22*	0.15*	0.12*	0.06*	0.03	0.09*	0.09
No Proper noun	Pearson	**0.38***	0.35	0.2	0.21	0.33	0.2	0.11	0.09	0.11
	Kendall Tau	**0.23***	0.21*	0.1*	0.15*	0.18*	0.06	0.08	0.12*	0.11*

to the top-performing baseline method i.e., maxSCQ, one can conclude that we performed 76% and 125% better on Pearson Rho and Kendall Tau for queries containing proper nouns while we only made 15% and 27% improvement in queries with no proper nouns.

5. Discussion

The idea of our work has been to explore how the geometric properties of neural embeddings can be exploited to define and estimate term specificity and use such estimates to predict query performance. The motivation is that neural embeddings of terms allow for estimating inter-term associations and determining terms' characteristics such as term specificity, which has shown to be correlated with the query difficulty (He & Ounis, 2004).

While it is not possible to measure frequency information from neural embeddings, they are convenient for identifying the set of highly similar terms to a given term based on the terms' vector representation in the embedding space. We formalized our recursive definition of specificity, i.e., the specificity of a term can be determined from the context created by the surrounding highly similar terms within the neural embedding space. In order to formalize specificity, we defined the notion of the *ego − network* of a term within the embedding space. Such a network provides a context for defining specificity-based QPP metrics. Based on the *ego − network* and terms in the neighborhood of the ego term, we proposed three different categories of specificity metrics including neighborhood-based, graph-based and cluster-based. Further, we utilized learning to rank strategy for measuring term specificity based on the collection of unsupervised specificity metrics in a supervised manner.

Furthermore, we provided two test collections based on two different well-established hierarchies, i.e., Wikipedia and DMOZ, to evaluate the proposed specificity metrics directly. The two test collections consist of 5-element long paths ordered from the most generic to the most specific term. We assessed the performance of the specificity metrics by ranking the terms on each path based on the proposed specificity metric and compared them to the actual ranking of the terms using the Kendall Tau rank correlation.

We demonstrated that it is possible to devise metrics, based on the pre-trained neural embedding-based representation of terms to perform pre-retrieval QPP. It is worth noting that since we rely on pre-trained word embeddings to extract the embedding vectors of query terms, our work can be impacted by the out of vocabulary problem if the query term is not included in the embedding collection. One remedy for the out of vocabulary problem would be to use methods that would estimate an embedding representation for unobserved terms (Hu, Chen, Chang, & Sun, 2019).

We showed that our graph-based metrics that estimates specificity of a query term based on its *ego − network* can lead to stronger performance on QPP when compared to the state of the art that considers term clusters based on neural embeddings (Roy et al., 2019). In addition, while there is no pre-retrieval method which performs consistently well among all the TREC topics, Closeness Centrality (CC), Degree Centrality (DC) and Inverse Edge Frequency (IEF) showed a fairly balanced performance over different topics sets and corpora.

Summing up all the experiments done on evaluating specificity-based pre-retrieval QPP methods, we can conclude that Closeness Centrality (CC) and Inverse Edge Frequency (IEF), which measure the importance of a vertex in a graph, on *ego − network*, are suitable indicators of specificity.

In summary, our key findings include:

- Pre-trained, corpus-independent neural embedding representation of terms enables competitive estimation of term specificity;
- Graph-based metrics (e.g. CC and IEF) that consider both the relationship among the ego node and its neighbors and the neighbors themselves have the best performance when they are evaluated directly via well-established hierarchies or in the context of pre-retrieval QPP application;
- Among graph-based metrics, those metrics that focus more on the influence of the ego term (node-influence metrics) lead to more accurate specificity metrics compared to those metrics metrics that consider all the nodes uniformly.

6. Conclusion and future work

In this paper, we proposed specificity-based pre-retrieval QPP metrics which calculate specificity of terms based on their neural embeddings. Our intuition is that the specificity of a term can be determined from the context created by the surrounding highly similar terms within the neural embedding space. Therefore, we have proposed three different categories of specificity metrics: neighborhood-based, graph-based and cluster-based. In our experiments, we first evaluated the performance of term specificity metrics using two different well-established hierarchies that include the actual ranking of the terms based on their specificity. Then, we compared our proposed specificity-based pre-retrieval QPP methods to the state-of-the-art baselines and showed that our graph-based metrics outperform baselines in terms of Kendall's τ and Pearson's ρ coefficient.

As future work, we intend to pursue the following tasks:

- Post-retrieval QPP: It has already been shown that utilizing neural embeddings of query terms is able to enhance the performance of post-retrieval QPP methods (Roy et al., 2019; Zamani, Croft, & Culpepper, 2018). Thus, we intend to propose post-retrieval QPP methods by expanding our embedding-based specificity metrics to measure document specificity in addition to query specificity. Based on the idea that 'the more specific the list of retrieved documents are, the easier the query would be', we postulate that by integrating the specificity of the query with the specificity of the retrieved documents, one would be able to perform stronger post-retrieval query performance prediction.
- Collection difficulty: Instead of estimating query difficulty, we can incorporate neural embeddings in order to measure the

collection difficulty. Same as query difficulty estimation, this can be beneficial to predict the retrieval performance.
- Specificity metrics applications: we intend to leverage the proposed specificity metrics in different applications. For instance, to expand the query by more specific terms in order to enhance the retrieval performance or to recommend more specific entities to users on social media in order to improve personalized recommendation.

CRediT authorship contribution statement

Negar Arabzadeh: Conceptualization, Software, Validation, Investigation, Formal analysis, Writing - original draft. **Fattane Zarrinkalam:** Methodology, Investigation, Project administration, Writing - review & editing. **Jelena Jovanovic:** Conceptualization, Methodology, Writing - review & editing. **Feras Al-Obeidat:** Funding acquisition, Conceptualization, Validation. **Ebrahim Bagheri:** Funding acquisition, Conceptualization, Supervision, Methodology, Software, Writing - review & editing.

References

Arabzadeh, N., Zarrinkalam, F., Jovanovic, J., & Bagheri, E. (2019). *Geometric estimation of specificity within embedding spaces. Proceedings of the 28th ACM international conference on information and knowledge management, CIKM 2019, Beijing, China, November 3–7, 2019*2109–2112. https://doi.org/10.1145/3357384.3358152.

Benz, D., Körner, C., Hotho, A., Stumme, G., & Strohmaier, M. (2011). *One tag to bind them all: Measuring term abstractness in social metadata. The semanic web: Research and applications - 8th extended semantic web conference, ESWC 2011, Heraklion, Crete, Greece, May 29 - June 2, 2011, proceedings, part II*360–374. https://doi.org/10.1007/978-3-642-21064-8_25.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics, 5*, 135–146.

Carmel, D., & Yom-Tov, E. (2010). *Estimating the query difficulty for information retrieval. Proceeding of the 33rd international ACM SIGIR conference on research and development in information retrieval, SIGIR 2010, Geneva, Switzerland, July 19–23, 2010*911. https://doi.org/10.1145/1835449.1835683.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

Hashemi, H., Aliannejadi, M., Zamani, H., & Croft, W. B. (2019a). Antique: A non-factoid question answering benchmark. arXiv:1905.08957.

Hashemi, H., Zamani, H., & Croft, W. B. (Zamani, Croft, 2019b). *Performance prediction for non-factoid question answering. Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval*55–58.

Hauff, C. (2010). Predicting the effectiveness of queries and retrieval systems. *SIGIR Forum, 44*(1), 88. https://doi.org/10.1145/1842890.1842906.

Hauff, C., Hiemstra, D., & de Jong, F. (2008). *A survey of pre-retrieval query performance predictors. Proceedings of the 17th ACM conference on information and knowledge management, CIKM 2008, Napa Valley, California, USA, October 26–30, 2008*1419–1420. https://doi.org/10.1145/1458082.1458311.

Hauff, C., Kelly, D., & Azzopardi, L. (2010). *A comparison of user and system query performance predictions. Proceedings of the 19th ACM conference on information and knowledge management, CIKM 2010, Toronto, Ontario, Canada, October 26–30, 2010*979–988. https://doi.org/10.1145/1871437.1871562.

He, B., & Ounis, I. (2004). *Inferring query performance using pre-retrieval predictors. String processing and information retrieval, 11th international conference, SPIRE 2004, Padova, Italy, October 5–8, 2004, proceedings*43–54. https://doi.org/10.1007/978-3-540-30213-1_5.

He, J., Larson, M., & de Rijke, M. (2008). *Using coherence-based measures to predict query difficulty. Advances in information retrieval, 30th European conference on IR research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings*689–694. https://doi.org/10.1007/978-3-540-78646-7_80.

Hu, Z., Chen, T., Chang, K.-W., & Sun, Y. (2019). Few-shot representation learning for out-of-vocabulary words. arXiv:1907.00505.

Jones, K. S. (2004). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 60*(5), 493–502. https://doi.org/10.1108/00220410410560573.

Kammann, R., & Streeter, L. (1971). Two meanings of word abstractness. *Journal of Verbal Learning and Verbal Behavior, 10*(3), 303–306.

Kapanipathi, P., Jain, P., Venkatramani, C., & Sheth, A. P. (2014). *User interests identification on Twitter using a hierarchical knowledge base. The semantic web: Trends and challenges - 11th international conference, ESWC 2014, Anissaras, Crete, Greece, May 25–29, 2014. Proceedings*99–113. https://doi.org/10.1007/978-3-319-07443-6_8.

Kwok, K. L. (1996). *A new method of weighting query terms for ad-hoc retrieval. Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR'96, August 18–22, 1996, Zurich, Switzerland (special issue of the SIGIR forum)*187–195. https://doi.org/10.1145/243199.243266.

Li, Y., Zheng, R., Tian, T., Hu, Z., Iyer, R., & Sycara, K. P. (2016). *Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. COLING 2016, 26th International conference on computational linguistics, proceedings of the conference: Technical papers, December 11–16, 2016, Osaka, Japan*2678–2688.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (Chen, Corrado, Dean, 2013a). *Efficient estimation of word representations in vector space. 1st International conference on learning representations, ICLR 2013, Scottsdale, Arizona, USA, May 2–4, 2013, workshop track proceedings*.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2018). *Advances in pre-training distributed word representations. Proceedings of the eleventh international conference on language resources and evaluation, LREC 2018, Miyazaki, Japan, May 7–12, 2018.*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (Sutskever, Chen, Corrado, Dean, 2013b). *Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems 26: 27th annual conference on neural information processing systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States*3111–3119.

Mimno, D. M., & Thompson, L. (2017). *The strange geometry of skip-gram with negative sampling. Proceedings of the 2017 conference on empirical methods in natural language processing, EMNLP 2017, Copenhagen, Denmark, September 9–11, 2017*2873–2878.

Mothe, J., & Tanguy, L. (2005). *Linguistic features to predict query difficulty. ACM SIGIR 2005 workshop on predicting query difficulty - methods and applications,*.

Oosterhuis, H., & de Rijke, M. (2018). *Differentiable unbiased online learning to rank. Proceedings of the 27th ACM international conference on information and knowledge management, CIKM 2018, Torino, Italy, October 22–26, 2018*1293–1302. https://doi.org/10.1145/3269206.3271686.

Orlandi, F., Kapanipathi, P., Sheth, A., & Passant, A. (2013). *Characterising concepts of interest leveraging linked data and the social web. Proceedings of the 2013 IEEE/WIC/ACM international joint conferences on web intelligence (WI) and intelligent agent technologies (IAT)-volume 01*. IEEE Computer Society519–526.

Roy, D., Ganguly, D., Mitra, M., & Jones, G. J. F. (2019). Estimating gaussian mixture models in the local neighbourhood of embedded word vectors for query performance prediction. *Information Processing and Management, 56*(3), 1026–1045. https://doi.org/10.1016/j.ipm.2018.10.009.

Segarra, S., & Ribeiro, A. (2016). Stability and continuity of centrality measures in weighted graphs. *IEEE Transactions on Signal Processing, 64*(3), 543–555. https://doi.org/10.1109/TSP.2015.2486740.

Thomas, P., Scholer, F., Bailey, P., & Moffat, A. (2017). *Tasks, queries, and rankers in pre-retrieval performance prediction. Proceedings of the 22nd australasian document computing symposium, ADCS 2017, Brisbane, Qld, Australia, December 7–8, 2017*11:1–11:4. https://doi.org/10.1145/3166072.3166079.

Voorhees, E. M. (2005a). *Overview of the TREC 2005 robust retrieval track. Proceedings of the fourteenth text retrieval conference, TREC 2005, Gaithersburg, Maryland, USA, November 15–18, 2005*.

Voorhees, E. M. (2005b). The TREC robust retrieval track. *SIGIR Forum, 39*(1), 11–20. https://doi.org/10.1145/1067268.1067272.

Zamani, H., Croft, W. B., & Culpepper, J. S. (2018). *Neural query performance prediction using weak supervision from multiple signals. The 41st international ACM SIGIR conference on research & development in information retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08–12, 2018*105–114. https://doi.org/10.1145/3209978.3210041.

Zarrinkalam, F., Fani, H., Bagheri, E., & Kahani, M. (2017). *Predicting users' future interests on twitter. Advances in information retrieval - 39th European conference on IR research, ECIR 2017, Aberdeen, UK, April 8–13, 2017, proceedings*464–476. https://doi.org/10.1007/978-3-319-56608-5_36.

Zhang, S., & Balog, K. (2018). *Ad hoc table retrieval using semantic similarity. Proceedings of the 2018 world wide web conference on World Wide Web, WWW 2018, Lyon, France, April 23–27, 2018*1553–1562. https://doi.org/10.1145/3178876.3186067.

Zhao, Y., Scholer, F., & Tsegay, Y. (2008). *Effective pre-retrieval query performance prediction using similarity and variability evidence. Advances in information retrieval, 30th European conference on IR research, ECIR 2008, Glasgow, UK, March 30-april 3, 2008. Proceedings*52–64. https://doi.org/10.1007/978-3-540-78646-7_8.