Estimating Gaussian Mixture Models in the Local Neighbourhood of Embedded Word Vectors for Query Performance Prediction

Dwaipayan Roy[a,*], Debasis Ganguly[b], Mandar Mitra[a], Gareth J.F. Jones[c]

[a] *Indian Statistical Institute, Kolkata, India*
[b] *IBM Research, Dublin, Ireland*
[c] *Dublin City University, Dublin, Ireland*

Abstract

The study of query performance prediction (QPP) in information retrieval (IR) aims to predict retrieval effectiveness. The specificity of the underlying information need of a query often determines how effectively can a search engine retrieve relevant documents at top ranks. The presence of ambiguous terms makes a query less specific to the sought information need, which in turn may degrade IR effectiveness. In this paper, we propose a novel word embedding based pre-retrieval feature which measures the ambiguity of each query term by estimating how many 'senses' each word is associated with. Assuming each sense roughly corresponds to a Gaussian mixture component, our proposed generative model first estimates a Gaussian mixture model (GMM) from the word vectors that are most similar to the given query terms. We then use the posterior probabilities of generating the query terms themselves from this estimated GMM in order to quantify the ambiguity of the query. Previous studies have shown that post-retrieval QPP approaches often outperform pre-retrieval ones because they use additional information from the top ranked documents. To achieve the best of both worlds, we formalize a linear combination of our proposed GMM based pre-retrieval predictor with NQC, a state-of-the-art post-retrieval QPP. Our experiments on the TREC benchmark news and web collections demonstrate that our proposed hybrid QPP approach (in linear combination with NQC) significantly outperforms a range of other existing pre-retrieval approaches in combination with NQC used as baselines.

Keywords:
Information Retrieval, Query Performance Prediction, Word Embedding

1. Introduction

Query performance prediction (QPP) is the task of automatically estimating (without any human input) the quality of the search results for a query. If a query is predicted to be "difficult", i.e., the search results are estimated to be of poor-quality, this prediction may be used to selectively suggest further actions, such as query reformulation or relevance feedback, in an attempt to improve retrieval performance. Of course, retrieval quality depends on a very large number of factors. Predicting query performance, or equivalently, estimating the difficulty of a query, is therefore a challenging problem.

Previously, researchers have sought to identify features that may indicate query difficulty. For example, a query may be difficult because it is ambiguous. Classic examples of ambiguous queries include 'python' (with the programming language or the snake as possible senses) and 'Paris Hilton' (referring to either the celebrity or the Hilton group of hotels in Paris). Although such queries are ambiguous, the underlying information need in the mind of a user executing the query typically pertains to one of the possible senses

*Corresponding author.
Email addresses: dwaipayan_r@isical.ac.in (Dwaipayan Roy), debasis.ganguly1@ie.ibm.com (Debasis Ganguly), mandar@isical.ac.in (Mandar Mitra), gareth.jones@dcu.ie (Gareth J.F. Jones)

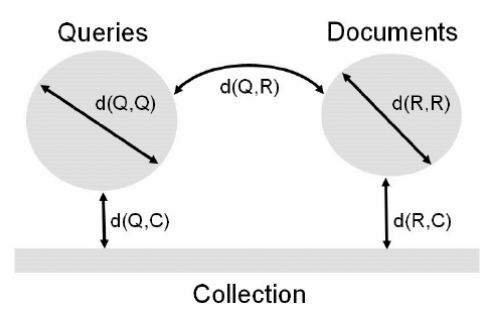

Figure 1: Broad classes of factors on which retrieval effectiveness usually depends (reproduced from [1]).

of the query. The top ranked documents retrieved by an information retrieval (IR) system may contain documents relating to different possible senses of the query. However, only a few among these documents, i.e., the ones which are related to the intended sense of the query, will be relevant. This can lead to poor IR effectiveness manifested in user frustration or confusion.

The difficulty of satisfying a user's information need in response to the query issued can also be related to the content of the collection being searched. For example, the ambiguity of a query can depend on the heterogeneity of the document collection on which the search is performed. If a user submits the query 'python' with the programming language sense of the term in mind, and the target collection consists of documents from the 'Computer Science' domain only, search results could well be perfectly satisfactory.

Thus, the factors that affect retrieval performance for a query may be related to the actual expression of the user's information need (e.g., query term ambiguity), or to the properties of the target collection (e.g., heterogeneity). The study in [1] presents a general model for the broad classes of factors affecting retrieval performance. This model is shown in Figure 1 (reproduced from [1]). In the figure, Q represents a set of queries corresponding to an information need, R the set of relevant documents, C the target document collection, whereas $d(\cdot\,,\cdot)$ represents various distance measures that may be interpreted as follows.

1. $d(Q, Q)$ represents query specificity. Intuitively, the space of possible queries corresponding to a specific and well-defined information need is relatively narrow. This corresponds to a small value of $d(Q, Q)$. On the other hand, a vague or under-specified information need may be represented by diverse queries, which result in a larger value of $d(Q, Q)$. Thus, a high value of $d(Q, Q)$ suggests that retrieval results may be unsatisfactory.
2. $d(Q, C)$ measures how discriminative the terms in query Q are with respect to the collection C. The presence of informative (discriminative) terms in the query is likely to yield high retrieval effectiveness.
3. $d(R, C)$ estimates how well-separated the relevant documents are from the rest of the collection. A high value is indicative of potentially effective retrieval performance.
4. $d(R, R)$ is a measure of the topical diversity of the relevant information. A high value indicates that relevant documents are diverse, and that a system may find it difficult to retrieve all the different aspects of the relevant information.
5. $d(Q, R)$ roughly corresponds to the semantic distance between the expression of the information need and the relevant content. A high value suggests that there may be a significant vocabulary gap between a query and its relevant documents. For such queries, retrieval quality is likely to be poor.

Out of the factors affecting query performance enumerated above, our work in this paper focuses on the first one, namely $d(Q, Q)$ or *query specificity*. Query specificity is traditionally estimated by aggregating various collection-level statistics over the query terms (and possibly other terms from the top ranked documents). Examples of such collection statistics include the average or maximum of the inverse document frequency (IDF) values of query terms [2]. The rationale behind these approaches is that terms with high IDF values are relatively rare in the collection, leading an IR model to easily distinguish the documents containing these terms from the rest of the collection.

Query specificity is also inversely related to the ambiguity of query terms. Thus, some existing QPP approaches attempt to quantify query ambiguity in order to estimate query difficulty [2]. These approaches generally use static resources such as WordNet [3]. Specifically in studies such as [4, 2], an aggregate of the number of possible senses of the query terms from WordNet has been used as a measure of query ambiguity. The limitation of such WordNet-based approaches is that they are unable to capture collection-specific semantic relationships between query terms. For example, it is not possible for a WordNet-based approach to predict that the word 'Hilton' may be less ambiguous in a collection of hotel reviews, as a result of which the query 'Paris Hilton' is less likely to be difficult in such a collection.

In our work, we propose to estimate $d(Q, Q)$ by leveraging the vector embedding of words, which potentially captures collection specific term-semantics. The embedded space of word vectors is then used to quantify the ambiguity of a query. Specifically speaking, we assume that each sense of a query term roughly corresponds to a cluster of word vectors around the neighbourhood of the query term vector. Intuitively speaking, the number of clusters around the neighborhood of a query term is thus a potential indicator of the specificity of the term, i.e. lower the number of clusters, higher is the specificity of the term.

Quantifying $d(Q, Q)$ based on this idea of embedded word vectors has an advantage over approaches that use static resources such as WordNet. The word embedding based approach can dynamically adapt itself according to the diversity of the collection. For example, the word vector representations, being collection specific, can potentially identify that 'Paris Hilton' is not a difficult query for a collection of hotel reviews. In such a collection, it is expected that the neighbouring vectors of the embedding for 'Hilton' would correspond to the hotel chain instead of the celebrity.

The rest of this paper is organized as follows. Section 2 highlights our research contributions. In Section 3, we start by reviewing existing literature on query performance prediction and differentiate our work from existing approaches. Our proposed method is based on word embedding. Hence, later in the section, we briefly describe word embedding techniques, focusing on a particular algorithm named `word2vec`. In Section 4, we formally describe our word-embedding based predictor. The evaluation setup and the results of our experiments with the proposed method are presented in Section 5 and 6, respectively. Finally, in Section 7, we conclude the paper with directions for future work.

2. Our Research Contributions

Research Objective. The objective of our research is to improve the state-of-the-art query performance prediction (QPP) effectiveness. Devising effective QPP is a step towards developing smarter search systems that, in principle, would be able to carry out alternate actions for difficult search queries, including asking the user a list of possible query reformulations, presenting different facets of search results, or recommending documents that other people have found useful on related topics.

Theoretical Contribution. The main theoretical contribution of the paper is to propose a generative model of query specificity, namely estimating $d(Q, Q)$, based on the application of embedded word vectors. We would like to emphasize that the application of word vectors for QPP is novel.

Practical Implications. The practical implication of the paper lies in empirically validating that our proposed QPP method significantly outperforms a wide range of other existing QPP approaches on a number of standard benchmark datasets. Our proposed method is able to achieve gains of more than 10% in correlation with human assessed query difficulty.

3. Background and Related Work

Predicting the performance of a query has been an active research area in the IR community over the last decade with many positive outcomes. According to the broad classes of features useful for QPP (see Figure 1), generally speaking, QPP approaches can be divided into two categories - a) *pre-retrieval* approaches, which only make use of $d(Q, Q)$ and $d(Q, C)$; and b) *post-retrieval* approaches, which include the other three. We start this section with a somewhat formal introduction to QPP in Section 3.1, and then provide a survey of QPP approaches belonging to the pre-retrieval and post-retrieval categories in Section 3.2 and Section 3.3 respectively.

3.1. Formal Description

Following the exposition of the study in [5], a query performance predictor $\mathcal{P}$ can be formally represented as a general combination of two predictor functions (pre-retrieval and post-retrieval), each of which outputs a score based on a given query, collection, external information and documents retrieved with an IR model. This is stated formally in Equation 1.

$$\mathcal{P} = g(\mathcal{P}_{pre}, \mathcal{P}_{post}),\ \mathcal{P}_{pre}, \mathcal{P}_{post} : (Q, C, E, R) \rightarrow \mathbb{R}, \tag{1}$$

where Q is a given query, C is the target corpus of documents, E is any external resource (e.g. Wikipedia and WordNet), and R denotes a retrieval function that returns a ranked list of documents from C in response to Q. Typically, the output of a predictor function, as shown in Equation 1, is interpreted as the specificity or clarity measure of a query. This implies that a query with a smaller value of a predictor function output is interpreted to be more difficult for a retrieval engine (in terms of effectively retrieving relevant documents from the collection) than a query with a larger predictor output.

Depending on how C, E or R are used, performance predictors can be divided into two categories [6].

- Pre-retrieval approaches, which only make use of query term statistics (e.g., specificity of the query terms, average number of senses from WordNet [3]) to estimate query difficulty [4, 2]. These methods do not have to rely on executing the query on an IR system to compute f_{pred}. For such predictors, Equation 1 can be rewritten as shown in Equation 2.

  $$\mathcal{P}_{pre} = f_{pred}(Q, C, E, \varnothing) \rightarrow \mathbb{R}. \tag{2}$$

  That is, R is set to $\varnothing$, as no retrieval function is used in these approaches; only the query terms (Q), collection statistics (C), and possibly some external resource(s) (E) are used to estimate query difficulty.
- Post-retrieval approaches, which can additionally harness information obtained from the top ranked documents retrieved for the given query (e.g score of top documents, robustness, and clarity of retrieval) [7, 1, 8, 9]. A post-retrieval predictor can be notated as shown in Equation 3.

  $$\mathcal{P}_{post} = f_{pred}(Q, C, E, R) \rightarrow \mathbb{R}. \tag{3}$$

  That is, in these approaches a list of documents retrieved by the method R is used in addition to the information used by pre-retrieval predictors.

In terms of the broad classes of distance measures between the different features useful for query performance prediction (schematically shown in Figure 1), all pre-retrieval predictors are based on some function of $d(Q, C)$ and $d(Q, Q)$. In contrast, a post-retrieval predictor is usually based on some function of $d(Q, R)$, $d(R, R)$ and $d(R, C)$ in addition to $d(Q, C)$ and $d(Q, Q)$.

3.2. Pre-retrieval Performance Predictors

Pre-retrieval performance predictors can be broadly classified into the following types according to the query term features used for computing the predictor function: a) *specificity*, b) *rank sensitivity*, c) *term relatedness*, and d) *ambiguity* [5]. Following is a brief description of each.

3.2.1. Specificity

Specificity based predictors make use of the assumption that discriminative terms are better able to extract relevant content from the whole collection. The predictor in this case is a function of $d(Q, C)$, i.e. how specific or discriminative are the query terms. Estimates of $d(Q, C)$ using existing approaches usually rely on both inverse document frequency (IDF) and inverse collection frequencies (ICTF) of terms. For example, the study in [7] computes the specificity of a query as the average of the IDF values over the constituent terms. In some studies, e.g. [10], the term with the maximum IDF ($MaxIDF$) is used to estimate query specificity. Some studies, e.g. [11], compute the standard deviation of IDF from the average of ICTF values ($AvgICTF$) as the specificity-based predictor function.

The work reported in [12] proposes a number of predictors that use the collection based specificity to predict query difficulty. More specifically, three predictors, which respectively employ: a) the sum of collection and each query term similarity ($SumSCQ$), b) normalized query and collection similarity ($AvgSCQ$), and c) the maximum of collection and query term similarity of the query term ($MaxSCQ$), were proposed as QPP approaches [13].

3.2.2. Rank Sensitivity

In *rank sensitivity* based predictors, proposed in [12], the intention is to estimate the difficulty of a query by term weight variation across the collection. It is reasoned that if the query term weight distribution is even across all the documents of the collection, it would be difficult to distinguish the relevant documents from those non-relevant documents containing the query terms [12]. In other words, the greater the deviation of the query term weights, the easier the query is to satisfy. The authors propose three methods based on this conjecture: a) the sum of query term weight variations ($SumVAR$), b) normalized sum of query term weight variations ($AvgVAR$), and c) maximum of query term variations ($MaxVAR$).

3.2.3. Term Relatedness

In contrast to the specificity and the rank sensitivity approaches that apply the estimates of $d(Q, C)$ values, the term relatedness approaches make use of the $d(Q, Q)$ estimates. More specifically, *term relatedness* based predictors, such as *AvLesk* [14], *AvPath* [15] and *AvVP* [16] use external resources (usually WordNet [3]) or co-occurrence statistics to determine the semantic distances between the query terms. The key idea behind using estimates of $d(Q, Q)$ is that semantically related queries are likely to be specific of a focused information need. On the other hand semantically unrelated query terms are likely to be indicative of queries that are underspecified.

One of the shortcomings of this predictor is that it cannot be applied for single term queries, for which the term relatedness value will be zero. Moreover, WordNet based semantic distances are static in nature and cannot be adapted according to the domain of the collection. Although co-occurrence statistics based approaches make provision for domain adaptation, they fail to take into account the local context of words (e.g. the local context of word embedding approaches). Our approach of computing $d(Q, Q)$ overcomes these shortcomings.

3.2.4. Ambiguity

The presence of an ambiguous term in the query is likely to make the information need imprecise. Since an ambiguous term has multiple senses associated with it and only one of them is likely to be relevant to the information need, it is likely that documents pertaining to each such sense may appear within the top retrieved list, as a result of which, it is difficult for a retrieval function to perform effectively [17]. It is thus reasonable to presume that an ambiguous query is usually a difficult one [6, 18].

Since our proposed QPP approach is based on the quantified ambiguity of the query, we will take a look at the predictors in this category more closely. The study [4] proposes a QPP approach (named $AvQC$), which involves computing the average inter-document similarity of all pairs of documents that contain at least one query term. A variant of AvQC, named $AvQCG$, is also proposed, which additionally includes document pairs containing all query terms in the average computation. A cosine similarity metric was used to calculate these similarities between document pairs. The rationale here is that the greater the inter-document similarity, the more coherent the documents are, which in turn implies an increase in the likelihood of the retrieval effectiveness [4].

A major drawback of this inter-document based QPP method is that it is not time scalable due to its quadratic time complexity of iterating over ${}^{N}C_{2}$ possible document pairs, where N is the number of documents containing at least one query term. For a large collection, a heuristic to overcome this problem is to sample a subset of document pairs for the computation [19].

Some existing approaches use WordNet [3] to find the ambiguity in a query [18]. In this proposed method, named Averaged Polysemy (AvP), all possible non overlapping combination of the query terms (up to word windows of length 5 are considered to check the number of senses associated with the query. The number

of senses found is then averaged over all terms considered. The predictor function is then defined to be inversely related to this estimated ambiguity value of the query. In another variant, named Averaged Noun Polysemy ($AvNP$), only the noun senses are considered, unlike all the senses used in AvP.

A performance comparison of these pre-retrieval predictors can be found in [2]. We refer the reader to [6, 5] for comprehensive details of alternative pre-retrieval performance predictors.

3.3. Post-retrieval Performance Predictors

In this section, we briefly describe some of the well known post-retrieval query performance predictors. Unlike pre-retrieval predictors, the performance of post-retrieval methods depends on the set of top-ranked documents (i.e. the set R of Equation 3 which in turn depends on the underlying retrieval model. As classified in [20], post-retrieval predictors can be divided into three types, based on the following criteria:

1. The Clarity (a measure of distinctiveness) of the top-retrieved documents compared to the whole collection [7, 21, 22].
2. Quantifying a notion of the robustness of the top retrieved documents [6, 23, 24, 25, 26].
3. A function (typically variance) of the retrieval scores of the top-ranked document list [9, 27, 8, 28].

Earlier QPP approaches, e.g. Clarity score (CS) [7], and Weighted Information Gain (WIG) [29]), involve computation of the estimation score with the help of the vocabulary overlap between the query Q, the top ranked documents R and the collection C (see Equation 3). The approach proposed in [30] applied relevance feedback information for performance predictions. Reference list based post-retrieval performance predictors are proposed in [20, 32, 33].

The Normalized Query Commitment (NQC) [34, 9] predictor measures the standard deviation of retrieval scores over the set R, i.e. the top documents retrieved for the query. The study in [9] reports that the variation within the retrieval scores has a correlation with the query performance. A high deviation is an indication that a retrieval system is able to effectively separate out the top k set from the rest of the ranked list, potentially indicating that the relevant and the non-relevant documents are well separated. On the other hand, a low variance indicates that it is difficult for a retrieval system to separate the top-k items on the basis of the Retrieval Status Values (similarity scores). In spite of being a simple and computationally fast approach, NQC has been reported to perform well for QPP, significantly outperforming the more computationally expensive predictors, e.g. the CS and WIG [9].

A somewhat similar approach (also based on variances of retrieval scores) is presented in [35]. Term overlap between multiple query expansion methods was used for QPP in [36, 37]. In recent studies, a wide range of supervised methods have been applied for the QPP task, e.g., the study in [38] applied learning to rank for effective estimation of the predictor functions. For a comprehensive study of post retrieval predictors can be found in [6, 5].

3.4. Combination of Predictors

It is common practice to combine different predictors together for more effective QPP. Pre-retrieval predictors are reported to be combined using linear regression [19]. However, Kurland et al. showed that a combination of pre-retrieval and post-retrieval predictors is more effective in practice [39]. Their reports that this combined approach significantly outperforms the stand-alone performance of both these predictors, and also outperforms a range of different pre-retrieval combinations. Similar observations are reported in [38, 23]. Formally speaking, a pre-retrieval and a post-retrieval predictor can be linearly combined as shown in Equation 4.

$$\mathcal{P}^{*}_{comb}(Q) = \alpha\mathcal{P}_{pre}(Q) + (1-\alpha)\mathcal{P}_{post}(Q) \tag{4}$$

In Equation 4, $\mathcal{P}_{pre}$ and $\mathcal{P}_{post}$ respectively denote pre-retrieval and post-retrieval performance predictors (Equation 2 and 3), and $\alpha \in [0, 1]$ denotes a linear combination parameter.

In the context of the related literature, the focus of our work is to develop a word embedding based pre-retrieval predictor function that uses the embedded word vectors in computing $d(Q, Q)$ (see Figure 1). We then seek to linearly combine our word embedding based pre-retrieval predictor with a standard post-retrieval prediction function, namely NQC.

3.5. Word Embedding

Word embedding algorithms, such as `word2vec` [40], `GloVe` [41]) and `fasttext` [42] represent words as real-valued vectors in an abstract space of a preset number of dimensions (typically in hundreds). Given a large amount of unlabeled training data, a word embedding algorithm, such as `word2vec`, learns a dense vector representation of words such that the similarity between a pair of word vectors reflects the semantic similarity between them. Vector space algebra in the space of embedded word vectors has been shown to empirically demonstrate analogy and composition effects. For example, [40] reports that

> "vec('Madrid') - vec('Spain') + vec('France') is closer to vec('Paris') than to any other word vector in the collection."

Word embedding and neural network based techniques have attracted significant interest in the natural language processing (NLP) community in recent times. In addition to their application in NLP ([43, 44, 45, 46]), these techniques are proving to be useful for many IR tasks as well. IR researchers have applied word embedding for different tasks, such as document ranking [47, 48, 49, 50, 51], query expansion [52, 53, 54], query classification [55], question answering [56] as well as similar item finding [57] or sponsored search [58] etc. For a comprehensive survey of applications of word embedding in IR, see [59].

To the best of our knowledge, word embedding techniques have not been used so far in an unsupervised manner to predict the performance of a query without human assessment. Utilizing the semantic similarity aspect of word embedding (that two words are embedded close to each other if they share contextual terms), in this work, we present a novel QPP method that is based on embedded vectors of words. In the context of our work, we focus only on the word vectors generated by the `word2vec` approach. Our propose method, however, does not rely only on the use of `word2vec`, and is general enough to work with other embedding approaches as well, such as `fastText` [42] and `GloVe` [41].

Some word embedding approaches aim to represent different senses of words as different vector representations, e.g. [60, 61, 62]. In the context of our work, sense embedding based approaches, such as `sense2vec`, are likely not to perform well because the senses that these embedding approaches represent are restricted to POS tags of words. In the context of our work, this may in fact be harmful if a multiple number of such senses correspond to the same POS tag, e.g. for the word 'cricket', both the game and the insect sense of the word are noun forms. Another problem with sense embedding based approaches is that the vector representations correspond to one particular sense of a term, which may lead to isolated neighbourhoods in the embedding space which may not be helpful to globally capture the true associations of a term to its possible senses. We empirically demonstrate this by employing `sense2vec` as one of our baselines.

In [63], the authors concluded that the structure of the local neighborhoods in semantic models carry useful semantic information regarding the different senses of a term, and that such topological properties can be used to analyze polysemy and for performing word sense inductions.

The study in [64] makes use of end-end deep learning in rank based approaches - both pointwise and pairwise, to predict query performance. The network in [64] was trained using millions of queries from the AOL query log along with the top documents retrieved from the ClueWeb. In contrast, we use a completely unsupervised approach for QPP. Similar to our proposed method of utilizing the word vectors around query terms, the study in [65] investigates using word vectors from top ranked documents to rerank the search results. In contrast to this work, we make use of word vectors trained over the whole document collection, instead of training word vectors over only the top-ranked documents.

4. Word Vector based Query Performance Prediction

In this section, we describe our proposed method for predicting query performance using word embeddings. Recall from Section 1 that our proposed approach is based on quantifying the ambiguity of a query. We first describe how embedded vectors can be leveraged to quantify query ambiguity (which is inversely related to the clarity or QPP score). Later in this section, we present our method for the combination of the estimated clarity (inverse of ambiguity) of a query with a post-retrieval predictor.

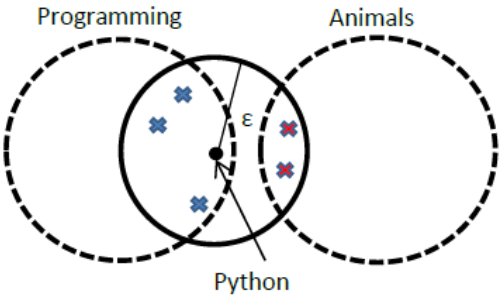

Figure 2: Illustrative diagram of the neighborhood of an ambiguous word with multiple senses.

4.1. Analysis of local neighbourhood in embedded space of word vectors

A word embedding algorithm, e.g., `word2vec` [40], maps each word w in a document collection into a real-valued vector $\mathbf{w} \in \mathbb{R}^d$ (d being an integer) in such a way that if two words are semantically related (in the sense that they occur in similar contexts), then their embedded vectors are also similar to each other (in terms of their inner product). The local neighbourhood around a word vector $\mathbf{w}$ thus comprises a set of words that are semantically related to it.

Now consider an ambiguous word such as 'python', which is associated with (at least) two very different senses - one related to the sense 'programming', and the other to that of 'animal'. Words associated with both these senses are expected to appear in the local neighbourhood around the vector for the word 'python'. Further, it is likely that the words in this neighbourhood will form two distinct clusters. This is because words in the neighbourhood associated with the 'animal' sense, such as 'snake', are likely to be of low similarity (high distance) with a word associated with the programming sense, e.g. 'jupyter'. In contrast, if each neighbouring term around the local neighbourhood of a word is similar (low distance) to every other term in the neighbourhood, it is likely that the word itself is unambiguous. This idea is schematically illustrated in Figure 2, which shows two distinct clusters in the ϵ-neighbourhood[1] of the word 'python' containing word vectors related to the 'programming' and the 'animal' sense shown with blue and red crosses respectively.

As discussed in Section 1, from a retrieval point of view, the ambiguity of a term is determined exclusively by its occurrences within the target corpus. For example, the term 'python' would most likely be unambiguous if the target collection consisted only of zoological reports. A static lexical resource such as WordNet [3] would still characterise such a term as ambiguous, based on the number of senses generally associated with the term. In contrast, when word embeddings are generated from the target document collection, our approach is expected to have the flexibility to capture the nature of terms based on their actual occurrences in the corpus.

4.2. Local neighbourhood as a Gaussian mixture model (GMM)

We now formalise the intuition outlined in the previous section into a probabilistic model that can be used to quantify the ambiguity of query terms. Let $\mathbf{q}$ be the vector corresponding to a query term $q \in Q$ (Q being the whole query comprised of a number of terms). Let $N_\epsilon(\mathbf{q})$ denote the ϵ-neighborhood of $\mathbf{q}$, defined as a in Equation 5.

$$N_\epsilon(\mathbf{q}) = \{\mathbf{x} : 0 \leq \cos^{-1}\left(\frac{\mathbf{x} \cdot \mathbf{q}}{|\mathbf{x}||\mathbf{q}|}\right) < \epsilon\} \tag{5}$$

Some observations worth noting in relation to Equation 5 are as follows. Firstly, we use the more intuitive notion of small *distances* around a pivot point to define a *neighbourhood* rather than the equivalent of maximizing the *similarity* values over a threshold as used in [66].

Secondly, since we work with distances instead of similarities, the most useful notion of distance in the embedded space of word vectors is the inverse of the cosine similarity or the angle between a pair of

[1] The ϵ-neighbourhood of a word vector $\mathbf{w}$ comprises a set of word vectors that are at most ϵ distance from $\mathbf{w}$.

vectors (as seen in Equation 5), as per the inner-product objective function of `word2vec`. Note that the more conventional distance metric of L2 (Euclidean) distance is not applicable in a space of word vectors embedded by the `word2vec` algorithm.

Thirdly, note that in Equation 5, we discard the points $\mathbf{x}$s from the neighborhood for which the inner product $\mathbf{x} \cdot \mathbf{q}$ is negative even if the absolute values of these negative angles could be smaller than ϵ. This is because negative angles between a pair of vectors do not indicate high similarity (inner-product) between them. Another intuitive way to think about avoiding negative angles is to use the transformation $\theta' = 2\pi - \theta$ for $\theta < 0$, which indeed represents a large angular distance outside the neighbourhood selection interval $[0, \epsilon)$.

We then consider these word vectors in each $N_\epsilon(\mathbf{q})$ as observations drawn from a Gaussian Mixture Model (GMM) of K components. For $\mathbf{x} \in N_\epsilon(\mathbf{q})$, we can define $P(\mathbf{x}|\boldsymbol{\theta})$ as shown in Equation 6.

$$P(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{6}$$

where $\boldsymbol{\theta} = \{\theta_k\}_{k=1,\ldots,K}$ is the set of parameters for the K components comprising π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ (the prior probability of selecting the kth component, the mean vector, and the covariance matrix for component k respectively).

Each Gaussian component in the neighbourhood of a query term potentially corresponds to a sense of the query term. Given the set of neighbourhood word vectors of a query term vector $\mathbf{q}$, the parameter values for the K components (i.e. the values of π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, $k = 1, \ldots, K$) can be estimated by expectation maximization (EM).

4.3. Quantifcation of query ambiguity post GMM estimation

Figure 3 shows an example scenario with the ϵ-neighborhoods of two query terms, one for an ambiguous term (left) and the other for an unambiguous term (right). Each histogram alongside the ϵ-neighborhoods shows the absolute number of points sampled from each component[2], indicative of the prior distribution $\boldsymbol{\pi}$. A skewed $\boldsymbol{\pi}$ distribution indicates a higher likelihood of selecting the most prevalent sense of a term by uniform sampling from $\boldsymbol{\pi}$, as shown in the right side histogram of Figure 3. On the other hand, the left side histogram of Figure 3 shows a more uniform $\boldsymbol{\pi}$ distribution, which indicates that it is more unlikely to be able to select the most prevalent sense of a term.

Post EM computation of the GMM parameters, following the intuition presented in Section 4.1, the intention is to compute the probability of generating words from the most dominant sense of a query term, which potentially correlates well with the inherent query specificity or clarity and is inversely related to the query ambiguity. This probability is expected to be high in cases where the true number of components (senses) is small, i.e. the query is relatively less ambiguous. Consequently, in such cases, the priors for this small number of components are high, or in other words, the variance of the prior values is high. For example, the right-hand side histogram of Figure 3 shows an example scenario of a query term that is associated with one dominating sense, where it can be seen that the variance of the priors is high. We proceed as follows to compute the probability of generating words from the most dominant sense of a query term.

Let $m \in \{1 \ldots K\}$ denote the most likely sense of the term q, as shown in Equation 7.

$$m = \arg\max_{k=1}^{K} \pi_k. \tag{7}$$

For a query term q, we define $P_{clarity}(q)$ as the probability that the query term vector $\mathbf{q}$ is generated by the m-th component of the GMM, i.e., the component corresponding to the most likely sense of q. We call this probability $P_{clarity}$ because it is likely to be indicative of the specificity of a query term.

[2]For illustrative purposes, the figure plots the absolute number of words in each Gaussian component instead of the normalized probability values.

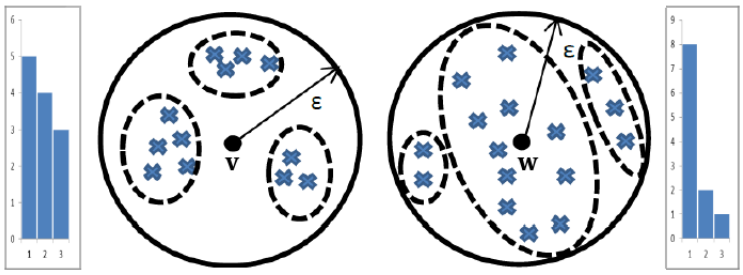

Figure 3: Illustrative diagram of the neighborhood of an ambiguous word with multiple senses.

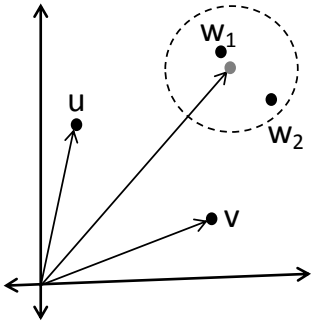

Figure 4: Word vector composition in an abstract two dimensional space (reproduced from [53]).

Speaking more precisely, we compute $P_{clarity}$ as shown in Equation 8.

$$\begin{aligned} P_{clarity}(q) =& \pi_m P(\mathbf{q}|\mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)) \\ =& \pi_m |2\pi\boldsymbol{\Sigma}_m|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}((\mathbf{q}-\boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1}(\mathbf{q}-\boldsymbol{\mu}_m)\right) \end{aligned} \tag{8}$$

A careful inspection of Equation 8 reveals that there are two factors on which the $P_{clarity}$ value depends. These are: i) π_m, which is the prior probability of choosing the most dominating sense of the query term q, and ii) $P(\mathbf{q}|\mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m))$, the posterior probability of sampling the query vector $\mathbf{q}$ from the selected component of the Gaussian mixture.

Informally speaking, the first component is high if the histogram of component membership of the estimated GMM is skewed. The second component informally measures how close a word is to the most dominating sense in the neighbourhood to the query tem q. More specifically, the value of this second component, i.e. $P(\mathbf{q}|\mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m))$, indicates how close the query term vector is to the mean vector $\boldsymbol{\mu}_m$ of the most prevalent sense. For an example, see the histogram on the right side of Figure 3 which indicates a high posterior likelihood of sampling $\mathbf{q}$ from the most dominating component (the one central to the neighborhood). On the other hand, this posterior probability is low on the left plot of Figure 3, where the query term vector $\mathbf{q}$ is relatively far away from the mean vector with the highest π_m value. The two components of Equation 8 together provides a predictor function for our QPP approach, which informally speaking, favours queries where neighbouring word vectors constitute a single dominating cluster and are close to this cluster centre.

The overall clarity of a multi-term query Q is then estimated by aggregating the individual probabilities for each query term as shown in Equation 9.

$$P_{clarity}(Q) = \prod_{q \in Q} P_{clarity}(q) \tag{9}$$

NN(gulf)	NN(war)	NN(gulf + war)
kuwait	fratricide	iraq
iraq	warfare	cold
mideast	wehrmacht	tonkin
groundfish	battlefield	fratricide
falkland	wartime	persian
sheikdom	strife	invasion
bahrain	battle	wartime
iraqi	conflict	afghanistan
hussein	fight	hussein
tonkin	postwar	superpower

Table 1: Neighbourhood of composed words often captures the phrasal effect.

4.4. Composing Query Term Vectors

For a multi-term query it is usually the case that each individual query term is associated with multiple senses, e.g., the word 'python', in isolation, may be associated with the 'animal' or the 'programming' sense. However, in conjunction with multiple query terms (e.g., *programming*), the underlying information need becomes more focused and less ambiguous. This is particularly true when the query terms constitute a phrasal concept (e.g., 'python programming'). Thus, if $\mathbf{w}$ is the addition of the word vectors $\mathbf{u}$ and $\mathbf{v}$, then $\mathbf{w}$ is likely to represent the concept corresponding to the composition of the words and is expected to be close to other words representing the same concept as u and v taken together, e.g. 'python' + 'programming' is expected to be close to 'jupyter'.

Figure 4 illustrates the idea of word composition in a two dimensional embedding space. We emphasize that this example is for visual illustration purposes only and does not exactly reflect the true geometry of a word embedding space, because firstly, word embedding spaces are of much higher dimensionality, and secondly, the distance metric in a word vector space is the angle between two vectors. In Figure 4, $\mathbf{w}$ represents the vector sum of $\mathbf{u}$ and $\mathbf{v}$. While $\mathbf{w}$ may not represent a word in the vocabulary, its neighbourhood is expected to contain embedding of words such as w_1 and w_2 (these words being a part of the vocabulary) that occur in similar contexts as u and v together. From Figure 4, we may also infer that the word w_1 is more similar to the composition than the word w_2, as $\mathbf{w_1}$ is closer to $\mathbf{w}$ than $\mathbf{w_2}$. The original paper introducing `word2vec` [40] uses an example of 'Russia' and 'river' being *close to* the word 'Volga' (equivalent to considering $\mathbf{w_1}$ in the neighborhood of $\mathbf{u} + \mathbf{v}$) in Figure 4).

Generally speaking, in the context of IR, word vector composition has been shown to improve the quality of relevance feedback [53]. Specifically speaking, in the context of our work, the composition between query terms may have a considerable effect in changing the neighborhood set, using which the mixture distribution is estimated.

Table 1 shows a concrete example from the TREC dataset, where it can be seen that the neighbourhood of the composition of the two words 'gulf' and 'war' better represents the relevant terms that could be useful for QPP. On the other hand, neighbourhoods of individual constituent query terms may contain words in their neighbourhoods that are not pertinent to the topic of the information need expressed by the query. Examples of such words in this specific example (as seen from Table 1) are 'groundfish' corresponding to the 'Gulf of Alaska', and general war related terms not related to the 'Gulf War' in particular, e.g. 'postwar', 'conflict' etc. This example shows the potential of using word composition for estimating the Gaussian mixture in our proposed QPP model.

To this end, in addition to aggregating GMM posteriors over individual query terms (as per Equation 9), we also include pairwise composition of successive query term vectors, i.e. $\mathbf{q'} = \mathbf{q_i} + \mathbf{q_{i+1}}$, in the aggregation process. Formally, for a given multi-term query with n word vectors, say $Q = \{\mathbf{q_i}\}_{i=1}^{n}$, we generate an 'expanded' query Q_e as shown in Equation 10.

$$Q_e = Q \cup \{\mathbf{q_i} + \mathbf{q_{i+1}}\}_{i=1}^{n-1}. \tag{10}$$

Document Collection	Collection Type	#Docs	Query Fields	Query Set	Query Ids	Dev Set	Test Set
TREC Disks 1 & 2	News	741,856	Title	TREC 2 ad-hoc	101-150	✓	
				TREC 3 ad-hoc	151-200		✓
TREC Disks 4 & 5 excluding CR	News	528,155	Title	TREC 6 ad-hoc	301-350	✓	
				TREC 7 ad-hoc	351-400		✓
				TREC 8 ad-hoc	401-450		✓
WT10G	Web	1,692,096	Title	TREC 9 Web	451-500	✓	
				TREC 10 Web	501-550		✓

Table 2: Dataset Overview

In the composition-based method, $P_{clarity}(q_i, q_{i+1})$ is then defined as per Equation 9 by using the vector sum of $\mathbf{q_i}$ and $\mathbf{q_{i+1}}$ in place of $\mathbf{q}$. We insert Q_e instead of Q in Equation (9) to aggregate contributions over query term vectors along with the composed ones. Note that we avoid considering all possible terms pairs $(\mathbf{q_i}, \mathbf{q_j})_{i \neq j}$ because successive term pairs are more likely to form phrasal concepts.

4.5. Combination with Post Retrieval Estimate

$P_{clarity}(Q)$, as defined in Equation 9, only considers the query terms for approximating the ambiguity (i.e. difficulty) of a query. However, it does not take into account the post-retrieval features that have been reported to be useful for performance prediction of a query [7, 1, 8, 9]. As stated in Section 3, the combination of pre and post retrieval methods can outperform both the individual predictors [23, 38, 39].

Following Equation 4 (see Section 3), the proposed predictor $P_{clarity}(Q)$ is combined with an effective post-retrieval QPP method, namely Normalized Query Commitment (NQC) [9].

NQC relies on the idea that a query is likely to yield better retrieval effectiveness if the Retrieval Status Values are skewed, or in other words, variance of the similarity scores is high. The final predictor is obtained after combining the proposed predictor with NQC using linear interpolation, as shown in Equation 11.

$$P^*_{clarity}(Q) = \alpha P_{clarity}(Q) + (1 - \alpha) NQC \tag{11}$$

The linear interpolation parameter α, in Equation 11, is the prior weight of selecting the embedding based pre-retrieval predictor. In our work, we train the parameter α on a development topic set, and apply the tuned value on the respective test topic set.

5. Experiment Setup

The objective of this study is to investigate whether word vector embedding of query terms can effectively capture query ambiguity, which in turn may be useful for predicting the retrieval performance of the query. To empirically validate our proposed method, we apply our proposed QPP method on a range of different benchmark test collections commonly used for ad hoc IR experiments. We start this section by describing the dataset characteristics and the baseline QPP methods. We then describe the tuning of the relevant QPP parameters.

We conduct our experiments on the standard TREC ad hoc tasks and web tasks datasets. Table 2 shows an overview of the details of these test collections. Indexing and retrieval on these collections are conducted using Lucene[3]. Stopword removal and stemming is known to positively influence the performance of retrieval. Hence, each word was stemmed using Porter Stemmer, and stopwords were removed using the SMART stopword list[4] before training the `word2vec` model.

[3] http://lucene.apache.org/core/

[4] http://www.lextek.com/manuals/onix/stopwords2.html

Predictor	Pre-retrieval Category			
	Specificity	Term relatedness	Rank Sensitivity	Ambiguity
SumSCQ [12]	✓			
AvgSCQ [12]	✓			
MaxSCQ [12]	✓			
AvIDF [7]	✓			
MaxIDF [10]	✓			
SumVar [12]		✓		
AvgVAR [12]		✓		
MaxVar [12]		✓		
AvLesk [14]			✓	
AvQC [67]				✓
AvQCG [67]				✓
AvP [18]				✓
AvNP [18]				✓

Table 3: Baseline Predictors Overview

5.1. Baseline pre-retrieval predictors

Since our proposed QPP approach is pre-retrieval based, for a fair comparison we choose as baselines the best pre-retrieval predictors as per the results reported in [5]. Table 3 enlists the baseline methods for our experiments (see Section 3 for a brief survey of these methods). All the QPP approaches in our experiments, i.e. the proposed method and the baselines, are implemented within the Lucene framework.

5.2. Neighbourhood size based baseline

To empirically demonstrate that it is useful to cluster the set of neighbouring words around a given query term for the purpose of estimating the number of senses associated with a term, we select a simple baseline, where we simply use the average number of terms found around the ϵ-neighbourhoods around the query terms. The intuition behind using this baseline is that a relatively isolated term is likely to be a polysemous because it is likely to be associated with multiple cluster centres (e.g. the point $\mathbf{v}$ in Figure 3). On the other hand, a term that is central to a concept is likely to be associated with a large number of words in its neighborhood (e.g. $\mathbf{w}$'s neighborhood in Figure 3).

Intuitively, one may think that the size of the ϵ-neighbourhood itself of the constituent query terms may act as a good predictor function, instead of more involved methods that require clustering of word vectors around the neighborhood. To investigate along this direction, we select a baseline, named $N_\epsilon(Q)$, that uses the average number of word vectors in the ϵ-neighborhood around the query terms as an indicator of query specificity.

5.3. Sense embedding based baseline

Instead of clustering the neighbourhood around query terms, one simple approach would be to use sense disambiguated word vectors to predict specificity of queries. To check if such a method would work well for QPP, we choose as a baseline a method that employs `sense2vec`[62].

The algorithm `sense2vec` learns embedding of each word-POS pair (in contrast to just a word only as in `word2vec`). For example, the embedding for the word 'bear' as a noun is different from its embedding as a verb. In effect, the `sense2vec` algorithm learns embedding of pseudo-words, where a pseudo-word is a concatenated representation of a word and its POS tag, e.g. 'bear:NOUN', 'bear:VERB' etc.

To investigate if `sense2vec` could be useful for QPP, we employ the following methodology as a baseline. To obtain nearest neighbors corresponding to each sense a query term we take into consideration a set of 4 frequent (and likely to be relevant for ranking documents) POS tags, namely 'nouns', 'verbs', 'adjectives'

and 'adverbs'. From an implementation perspective, we use the `sense2vec` API[5] that comes with pretrained POS tag annotated pseudo-word embedding available as a part of the API.

Next, similar to our proposed QPP algorithm, which computes how dominating is one sense over the others (see Equation 7), for this baseline method we compute

$$\text{QPP}_{s2v} = \frac{1}{|Q|} \sum_{q \in Q} \frac{\max_{t \in \mathcal{T}} sim(\mathbf{q_t}, N(\mathbf{q_t}))}{\sum_{t \in \mathcal{T}} sim(\mathbf{q_t}, N(\mathbf{q_t}))}, \tag{12}$$

where $\mathcal{T}$ denotes the set of four tags - nouns, verbs, adjectives and adverbs, $\mathbf{q_t}$ denotes the embedding of a pseudo-word (word:tag) combination of word q and tag t, $NN(\mathbf{q})$ denotes the nearest neighbor in terms of cosine distance (or in other words, the most similar neighbor to $\mathbf{q}$), and $sim(\mathbf{u}, \mathbf{v})$ denotes the cosine-similarity between two vectors $\mathbf{u}$ and $\mathbf{v}$.

Equation 12, in effect, computes how much does one sense (POS tag) of a query term dominates over the other ones. It is expected that for single sense terms, this ratio will be high, i.e. the similarity with the most similar neighbor of a query term corresponding to one particular sense will be considerably higher than the other ones. Note that our proposed QPP method also exploits a similar dominating effect with the help of the prior probabilities of the mixture components (as in Equation 7).

5.4. Combination with post-retrieval predictor

As discussed in Section 3, a combination of pre-retrieval and post-retrieval QPP approaches can outperform the ones which employ only the former or the latter. Consequently, in addition to using each standard pre-retrieval QPP based approach as a baseline, following the notion of Equation 4, we combine each pre-retrieval predictor a post-retrieval one. To enable fair comparison, we apply the same post-retrieval predictor in combination with different pre-retrieval approaches. In particular, we use NQC as the post-retrieval predictor.

We adopt the notation of denoting a combined predictor by inserting an asterisk over the corresponding pre-retrieval predictor's name. For example, MaxIDF* indicates the hybrid predictor that combines MaxIDF with the post-retrieval predictor NQC. The objective of the experiments is to show that our proposed QPP method combined with NQC, i.e., $P^*_{clarity}(Q)$ of Equation 11, can outperform NQC alone and other pre-retrieval predictors in combination with NQC. Note that, some of the baseline methods are based on measurement of ambiguity of query terms (see Table 3). For the hybrid methods, we reformulated clarity as unit difference from ambiguity and combined with the clarity based measurement NQC.

5.5. Linear combination of baselines

In order to empirically demonstrate that our proposed method is capable of achieving results that are beyond a simple combination of existing pre-retrieval predictors, we compare our proposed approach with an optimal combination of all of these approaches. Specifically, we introduce another baseline 'ALL' as the linear combination of the pre-retrieval baseline approaches enlisted in Table 3, as shown in Equation 13.

$$\text{ALL} = \beta_1 \text{SumSCQ} + \beta_2 \text{AvgSCQ} + \ldots \beta_{12} \text{AvP} + (1 - \sum_{i=1}^{12} \beta_i) \text{AvNP} \tag{13}$$

Since the optimization function of maximizing the rank correlation coefficients (see Section 5.6) is not differentiable, gradient descent approaches to find the optimal set of parameters are not applicable. Moreover, since the parameter space represents a 12 dimensional simplex, a complete grid search for the optimal set of parameters is also not feasible. To find the optimal parameter settings, we sample 1000 points at random with uniform probability from this 12 dimensional simplex of real values (some sampling algorithms with proofs of uniformity are presented in [68]), and choose the one yielding maximum rank correlation. Next, we combine the optimal linear combination of pre-retrieval based QPPs with the post-retrieval predictor - NQC to generate the baseline hybrid approach 'ALL_NQC'.

[5] `https://github.com/explosion/sense2vec`

5.6. Evaluation Metrics

We evaluate the QPP approaches in our experiments by measuring the correlation of the predicted ordering of the queries in each topic-set with the ground-truth ordering of the queries sorted by their average precision (AP) values. Following common practice in the IR community [9, 1, 2, 7, 4], to evaluate QPP methods, we use Pearson's ρ and Kendall's τ [69] as the rank correlation coefficient. Both these rank correlation methods reports a value in the range $\{1, -1\}$, higher values indicating better correlation.

5.7. Significance testing

We conduct statistical significance tests to examine whether our proposed method is significantly better than the best performing baseline method on the same test collection. More specifically, for a pair of queries (Q_i, Q_j), $(j > i$ to avoid self and duplicate comparisons), we define an indicator variable for a predictor function, which checks if the predicted ordering of a pair of queries (Q_i, Q_j) *conforms to* their true ordering, as shown in Equation 14.

$$\begin{aligned} \mathbb{I}_{f_{pred}}(Q_i, Q_j)) &= 1 \text{ if } f_{pred}(Q_i) > f_{pred}(Q_i) \wedge AP(Q_i) > AP(Q_j) \\ &= 0 \text{ otherwise,} \end{aligned} \tag{14}$$

where $AP(Q)$, denoting the average precision value of query Q, is computed by making use of the relevance judgments. We then perform the significance tests by comparing the indicator values for each query pair of two predictor functions - one for our method and the other for the best performing baseline.

5.8. Parameter Tuning

In order to compute the word vector based predictor function, word vectors were trained individually on each document collection (see Table 2). The dimensionality of the word vectors for estimating the $P_{clarity}(Q)$ scores was set to 200 using the *cbow* model of `word2vec` with negative sampling [40].

For our experiments, the underlying retrieval models employed to compute the NQC scores are Language Model with Jelinek Mercer smoothing (LM-JM) [13, 70], and BM25 [71]. As per standard settings in the reported literature [70], the LM-JM smoothing parameter (λ) was set to 0.4, whereas for BM25, k_1 and b, were set to the standard values of 1.2 and 0.75 respectively. The number of top ranked documents used to compute the NQC scores was set to 100 as per [9].

The GMM based predictor function relies on the ϵ-neighbourhood set of words for each query term. We set the value of $\epsilon = 0.1$ in our experiments, by varying its value within a range of $[0, 0.5]$ in our initial experimental investigation. Another parameter of the GMM-based predictor is the number of components (K) of the underlying GMM that is to be used to estimate the posterior of the observed vectors in the neighbourhood around a query term. A problem with selecting a constant value of K for all queries is that the number of terms in a query can vary considerably, making it less likely that a constant value of K will be effective for all queries. Instead, we make this parameter depend on the number of vectors that we find around the ϵ-neighbourhood. Note that for different query terms, the ϵ-neighbourhood can contain a variable number of terms. To select K, we define a parameter γ as shown in Equation 15. defined as

$$\gamma = \frac{|N_\epsilon(\mathbf{q})|}{K} \tag{15}$$

where γ represents the average number of words in each component of the GMM. This approach admits a variable number of components (K) in the GMM depending on the number of semantically similar words found in its ϵ-neighbourhood. Each component (sense) of the GMM comprises γ words on an average.

For the development topic sets, we varied γ in the range from 5 to 15 in unit steps. To simplify the GMM estimation, we compute $P_{clarity}(Q)$ (Equation 8) score of a query by considering a GMM with uniform prior (π) and unitary covariance matrix (Σ) which results in an isotropic Gaussian Mixture. Note that an isotropic unitary Gaussian Mixture with uniform priors is equivalent to K-Means clustering of the constituent points (word vectors) [72, Chapter 9].

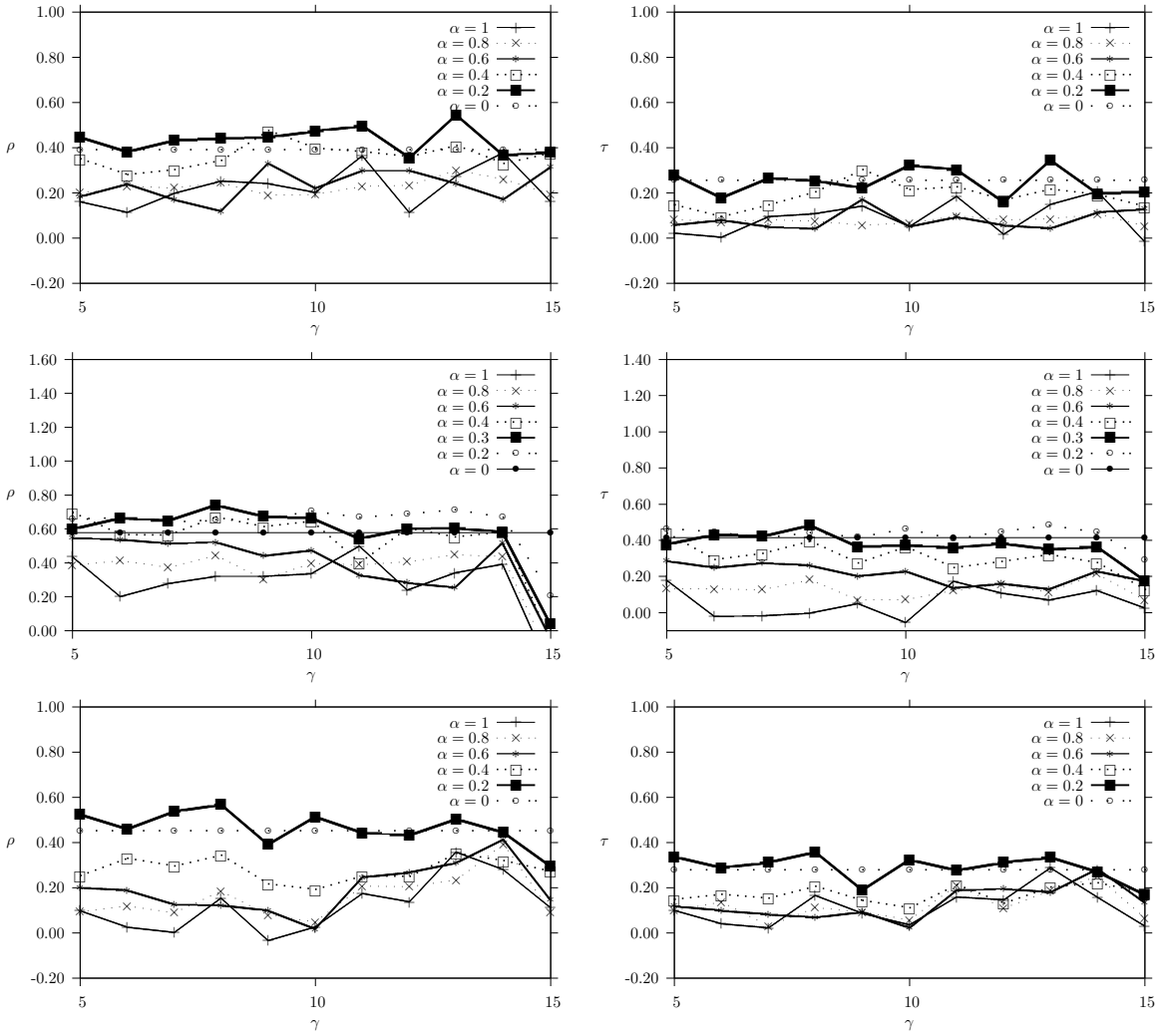

Figure 5: QPP sensitivity measured with Pearson's ρ (left) and Kendall's τ (right), for different values of α of Equation 11 and γ of Equation 15, on development topic sets TREC 2 (top), TREC 6 (middle) and TREC 9 (bottom).

Another parameter of our proposed method is the linear combination weight α (see Equation 11) which denotes the importance of the word embedding based predictor. The parameter α is varied in the range 0 to 1 in steps of 0.1. A value of $\alpha = 0$ degenerates the prediction measure $P^*_{clarity}(Q)$ to the NQC measure, whereas a value of $\alpha = 1$ solely uses the pre-retrieval word vector based prediction.

Both the parameters α and K are trained separately on each development topic set, i.e. TREC 2, TREC 6 and TREC 9 (see Table 2). The reason for using three different topic sets for tuning the parameters is due to the fact that the underlying document collections are different for these topic sets. The particular values of α and K that yield the optimal predictions on the development sets are then applied on the topics of the corresponding test sets. The interpolation parameters (α in Equation 4) for all hybrid QPP approaches are also tuned on the corresponding training topic sets, namely TREC 2, TREC 6 and TREC 9.

Method	TREC 2				TREC 6				TREC 9			
	Parameters		Correlation		Parameters		Correlation		Parameters		Correlation	
	α	γ	ρ	τ	α	γ	ρ	τ	α	γ	ρ	τ
MaxIDF*	0.1	N/A	0.3046	0.2212	0.1	N/A	0.7238	**0.4824**	0.1	N/A	0.2157	0.2336
AvgIDF*	0.1	N/A	0.3422	0.2473	0.1	N/A	**0.7645**	**0.4824**	0.1	N/A	0.4024	0.3675
SumSCQ*	0.1	N/A	0.0371	0.1478	0.1	N/A	-0.0817	0.0873	0.1	N/A	0.0595	0.0800
AvgSCQ*	0.1	N/A	0.3682	0.1771	0.1	N/A	0.2015	0.1788	0.1	N/A	0.3902	0.3348
MaxSCQ*	0.1	N/A	0.4288	0.3224	0.1	N/A	0.2463	0.2963	0.1	N/A	0.4443	**0.4198**
SumVAR*	0.1	N/A	0.1209	0.2065	0.1	N/A	0.2642	0.2424	0.1	N/A	0.1519	0.1339
AvgVAR*	0.1	N/A	0.4517	0.2408	0.1	N/A	0.5720	0.3992	0.1	N/A	0.3642	0.3299
MaxVAR*	0.1	N/A	0.4288	0.3224	0.1	N/A	0.5466	0.4449	0.1	N/A	0.3574	0.3463
AvP*	0.1	N/A	0.4465	0.2686	0.1	N/A	0.6283	0.4433	0.1	N/A	0.4359	0.2646
AvNP*	0.1	N/A	0.4238	0.2098	0.1	N/A	0.5765	0.4008	0.1	N/A	0.4920	0.2483
AvLesk*	0.2	N/A	0.4139	0.2522	0.2	N/A	0.6326	0.4727	0.2	N/A	0.3843	0.2401
AvQC*	0.1	N/A	0.4897	0.2898	0.1	N/A	0.3166	0.2686	0.1	N/A	0.2870	0.2441
AvQCG*	0.3	N/A	0.4896	0.2816	0.1	N/A	0.5263	0.3584	0.1	N/A	0.4311	0.2800
ALL_NQC	N/A	N/A	0.5297	0.3355	N/A	N/A	0.7266	0.4727	N/A	N/A	0.4539	0.3985
$N_\epsilon(Q)$	0.2	N/A	0.3027	0.2392	0.3	N/A	0.2093	0.2229	0.2	N/A	0.4517	0.2793
QPP_{s2v}	0.2	N/A	0.0853	0.0770	0.3	N/A	0.1094	0.0941	0.2	N/A	0.0907	0.0674
$P^*_{clarity}$ (w/o comp)	0.2	13	0.4790	0.2882	0.3	8	0.7320	0.4698	0.2	8	0.5273	0.3348
$P^*_{clarity}$ (with comp)	0.2	13	**0.5448**†	**0.3437**†	0.3	8	0.7417†	**0.4824**†	0.2	8	**0.5676**†	0.3577†

Table 4: Optimal parameter settings on the training topic sets for the hybrid approaches. A '†' in $P^*_{clarity}(Q)$ indicates that the performance is significantly better (with $p = 0.05$) than the best performing baseline method on the same dataset.

6. Results and Discussion

6.1. Parameter Tuning on Training Set

Figure 5 shows the QPP effectiveness obtained with different parameter settings of $P^*_{clarity}$ (both with and without compositions over query terms) on the development topic sets. The parameters varied were α (interpolation parameter for NQC combination) and γ (the average number of points in each cluster or GMM component). Optimal results on these topic sets are also summarized in Table 4.

The general trends observed from the plots of Figure 5 and the data reported in Table 4 are as follows. Firstly, the proposed hybrid approach $P^*_{clarity}(Q)$ outperforms NQC (corresponding to the line plots for $\alpha = 0$) for values of α in the range of $[0.2, 0.3]$. This indicates that the post-retrieval predictor NQC can be considerably improved by augmenting it with information from the semantic similarities between words, which a post-retrieval predictor alone cannot capture.

Secondly, from Table 4 it is observed that the combination of the word vector based approach with NQC outperforms combinations of NQC with all other existing pre-retrieval approaches, except for TREC 6 (where combination of NQC with AvgIDF is best). The likely reason for this is that the word vector approach utilizes term relationships with the help of the semantic similarities between the words, whereas the existing ones either treat each individual query term independently (e.g. AvgIDF*), or learn the semantic relationships in a static way (e.g. AvLesk*). Although AvgIDF* outperforms $P^*_{clarity}(Q)$ on TREC 6, its effectiveness for the other topic sets is not satisfactory. The effectiveness of $P^*_{clarity}(Q)$ is relatively stable and satisfactory across the topic sets.

Thirdly, it can be seen that the word embedding based QPP outperforms WordNet based QPP (AvLesk*, AvP* and AvNP*) because the word embedding based approach being trained on respective collections is better able to capture collection specific term semantics in comparison to the static semantics of a knowledge base.

Another important observation is that the simple baseline which uses the average number of words in the ϵ-neighbourhood of query terms, i.e. the baseline $N_\epsilon(Q)$, does not perform well in comparison to our

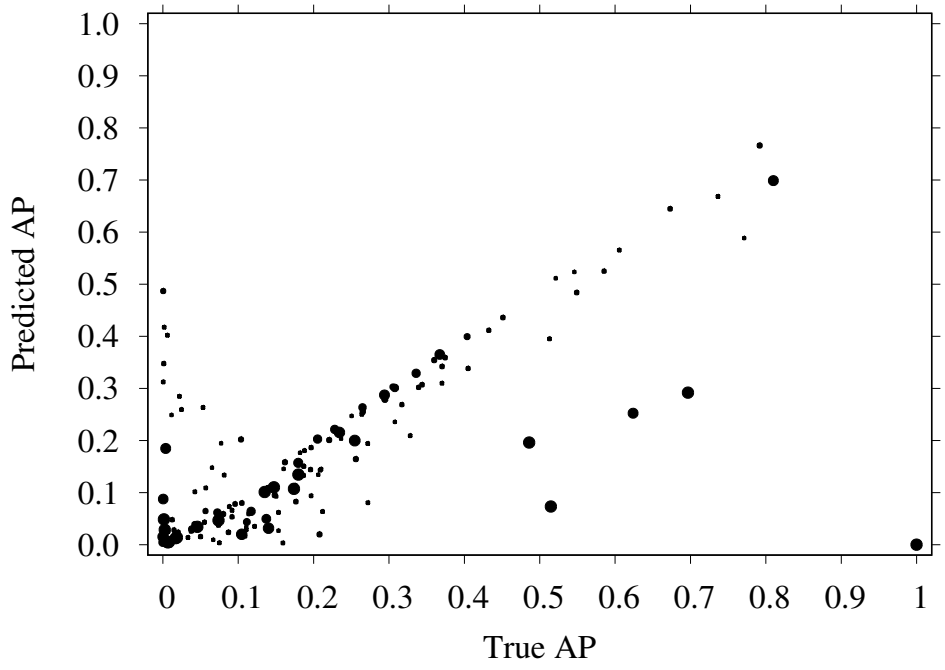

Figure 6: Per-query (comprised of training set queries, i.e. queries from TREC-2, TREC-6 and TREC-9) scatter plot of true AP vs predicted AP values. All predictions employed our proposed method with query term composition, i.e. the method $P^*_{clarity}$ (with composition). The radius of a circle for each query is indicative of the number of GMM components that yields optimal prediction.

proposed Gaussian estimation based predictor, which implies that a simple count of the number of nearest neighbours is not sufficient to produce a good quality QPP estimate. On the other hand, our proposed method is better able to predict the specificity of queries by clustering query terms into their possible senses.

Next, it is observed that the sense embedding baseline, QPP_{s2v}, performs poorly due to two main likely reasons. First, the very notion of 'sense' in sense embedding based approaches is restricted to part-of-speech (POS) tags which is rather a crude representation of the true sense of a term, For example, sense embedding based approaches are likely to introduce noise in the vector representations of the word 'cricket' because both the game and the insect sense of the word are noun forms. Secondly, sense embedding based approaches learn vector representations corresponding to one particular sense of a term, and in a way, this may lead to isolated neighborhoods in the embedding space which may not suffice to globally capture the true associations of terms with their possible senses exhibited in a large corpus.

Next, we observe that the linear combination of all pre-retrieval predictor approaches along with NQC, i.e. the method named 'ALL_NQC' yields better results than some (or all in the case of TREC 2) of the individual approaches. However, the combined approach is always worse than our proposed method, which shows that our proposed QPP method is able to achieve better results than a simple linear combination of existing QPP approaches.

Further, we note that the results with composition of the query terms are considerably better than without composition, which indicates that composition of the query term vectors plays a crucial role in estimating the specificity of a query. Since the composed version yields better results, henceforth in the paper, we report all experiments with the composed version only, and refer to it as $P*_{clarity}$ for simplicity.

	TREC 3		TREC 7		TREC 8		TREC 10	
	ρ	τ	ρ	τ	ρ	τ	ρ	τ
AvgIDF	0.2353	0.2490	0.4262	0.3796	0.5910	0.3518	0.3736	0.2359
MaxIDF	0.2285	0.2772	0.3524	0.2662	0.4938	0.2996	0.2219	0.1233
SumSCQ	-0.1289	-0.1380	0.0555	0.0612	0.0660	0.0612	0.2442	0.1886
AvgSCQ	0.3938	0.2963	0.4925	0.4302	0.4478	0.3159	0.2799	0.1886
MaxSCQ	0.3269	0.2107	0.4564	0.3949	0.5532	0.4384	0.3900	0.2947
SumVAR	0.0271	0.0008	0.2099	0.2522	0.5889	0.3976	0.3999	0.2637
AvgVAR	0.4042	0.3502	0.4372	0.3992	0.6475	0.4200	0.3585	0.2653
MaxVAR	0.3722	0.2655	0.4004	0.4018	0.6420	0.4371	0.4261	0.3159
AvP	0.1379	0.0567	0.3080	0.2776	0.2323	0.0994	0.0953	0.0215
AvNP	0.0290	-0.0190	0.3452	0.2975	0.2478	0.1623	0.0987	0.0261
AvgLesk	0.0297	-0.0375	0.2696	0.2509	0.0746	0.1014	0.2221	0.0896
AvQC	0.0920	0.0907	0.1032	0.0574	0.0560	0.0336	0.0472	0.0156
AvQCG	0.0932	0.0907	0.1135	0.0575	0.0563	0.0346	0.0473	0.0157
$P_{clarity}(Q)$	0.3222	0.2457	0.2043	0.2132	0.2132	0.1559	0.1544	0.0033
NQC	0.3146	0.1559	0.4580	0.3502	0.6717	0.4335	0.4676	0.2686
AvgIDF*	0.2782	0.2620	0.4551	0.4253	0.6315	0.3878	0.4119	0.2588
MaxIDF*	0.2862	0.3159	0.4204	0.3094	0.5815	0.3714	0.2157	0.2336
SumSCQ*	-0.1280	-0.1363	0.0602	0.0661	0.0728	0.0727	0.2455	0.1902
AvgSCQ*	0.4013	0.3061	0.5078	0.4351	0.4668	0.3241	0.2859	0.1967
MaxSCQ*	0.3347	0.2065	0.4706	0.4171	0.5739	0.4612	0.3945	0.2980
SumVAR*	0.0406	0.0155	0.2452	0.2718	0.6223	0.4253	0.4146	0.2718
AvgVAR*	0.4599	**0.3649**	0.4951	0.4433	0.6850	0.4543	0.4174	0.2996
MaxVAR*	0.4077	0.3127	0.4441	**0.4449**	0.6742	0.4659	0.4567	0.3518
AvP*	0.3129	0.1918	0.4784	0.4286	0.5221	0.2735	0.3281	0.1853
AvNP*	0.2327	0.1282	0.5141	0.4090	0.3697	0.1478	0.2634	0.1102
AvgLesk*	0.1794	0.1069	0.4095	0.3453	0.3457	0.3078	0.3292	0.1673
AvQC*	0.3224	0.1641	0.4047	0.2996	0.6393	0.4629	0.2870	0.2441
AvQCG*	0.3360	0.1771	0.4583	0.3518	0.6903	0.4335	0.4311	0.2800
ALL_NQC	-0.0516	-0.0580	0.3247	0.2441	0.4023	0.2784	0.3655	0.2800
$N_\epsilon(Q)$	0.3146	0.1560	0.4580	0.3502	0.6717	0.4335	0.4676	0.2685
$P^*_{clarity}(Q)$	**0.4741**†	0.3648†	**0.5362**†	0.4344†	**0.7070**†	**0.4798**†	**0.4936**†	**0.3524**†

Table 5: Comparisons of the word embedding based QPP method against various baselines on the test topic sets. NQC used LM-JM retrieval scores λ set to 0.4. A † in $P^*_{clarity}(Q)$ indicates the performance is significantly different than the best performing baseline methods.

6.2. Number of optimal clusters per query

A key hypothesis in our proposed method is that the neighbourhood of a word likely to be associated with multiple senses can, in principle, be better estimated with a relatively higher number of components in the mixture distribution (see Equation 6). To empirically verify this hypothesis, we investigate the relation of per-query QPP values, which can be considered to be the predicted average precision (AP) values, with the number of components in the GMM. Each point of Figure 6 shows the per-query true AP and the predicted AP values along the x and y axes respectively. The radius of each point is proportional to the number of GMM components producing the best prediction for that query (as per our empirical settings, the number of components was selected from the set of integers $\{5, 6, \ldots, 15\}$).

It can be seen from Figure 6 that for easy (high true AP) queries for which the predictor performs well (i.e. the system predicted AP values are high as well), a small number of GMM components yields optimal results. This is seen from the dots, most of which are of small radii, towards the top-right part of the figure. On the other hand, points with large radii values are frequently observed in the bottom-left part of the figure, indicative of the use of a large number of GMM components (corresponding to a relatively large

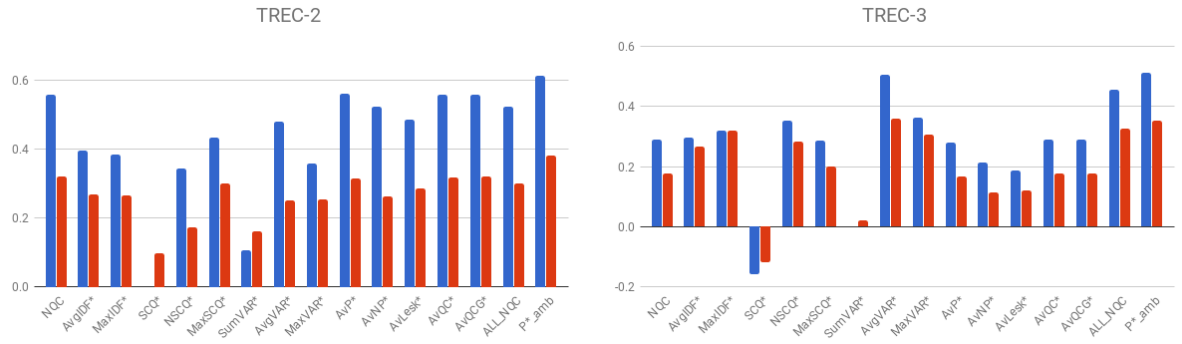

Figure 7: The performance of baseline methods and the proposed methods in terms of performance prediction of BM25 retrieval model on news collection topics TREC 2 and 3. In the plot, Y axis represents the correlation values obtained between the true AP of the retrieval model, and the prediction score. The blue bar and the red bar represent the Pearson's Rho (ρ) and Kendall's Tau (τ) correlation coefficient respectively.

number of possible senses) to capture the relatively high ambiguity (or low clarity) of these difficult (low true AP) queries. Indeed, this confirms our hypothesis that a large number of components works well for queries that lack specificity.

6.3. Performance on Test Set

After obtaining the optimal parameter settings on each development topic set, we applied these settings on its corresponding test set, the results are in Table 5. In this set of experiments, we test the performance of each pre-retrieval predictor used stand-alone and in combination with NQC (with LM-JM). We observe that the performance of the pre-retrieval approaches on their own (baselines and the word vector based one) are not satisfactory as can be seen from the upper half of Table 5.

Among the pre-retrieval only QPP methods, the average idf typically performs the best. This shows that the specificity of query terms is an important criteria for QPP, which our proposed word vector based method $P_{clarity}(Q)$ does not make use of. However, the word-vector based approach in combination with NQC, i.e. $P^{*}_{clarity}(Q)$, outperforms (in almost all cases) combinations of the specificity based predictors, such as MaxIDF and AvgIDF, with NQC. This shows that term semantics in combination with specificity information derived from the top retrieved documents can outperform approaches that do not use term semantics.

For TREC 3 and TREC 7 topic sets, AvgVAR* and MaxVAR* attain the best τ values respectively, the τ correlation values being close to our proposed method $P^{*}_{clarity}(Q)$. Compared to all the other baseline predictors, it can be observed that performance of $P^{*}_{clarity}$ is the most consistent.

Additionally, in order to investigate the performance of our proposed method with a different retrieval model based NQC scores, we report more results with BM25 based NQC scores in Figure 7, 8 and 9. For this set of experiments, we followed the same parameter tuning methodology, i.e. train on TREC 2, TREC 6 and TREC 9 topic sets and test on the respective test topic sets (see Table 2). As can be seen from the experiments on LM-JM retrieval model (also from [23, 39, 38]), the hybrid approach of pre-retrieval predictors with a post-retrieval one (in this case NQC) outperforms the effectiveness of stand-alone pre-retrieval methods, in this set of experiments investigating the effectiveness of our predictor with a different retrieval model based NQC scores (BM25 in this case), we only report evaluation measures for the hybrid combinations.

From Figure 7, 8 and 9, we can observe that the hybrid of the various QPP approaches with BM25 based NQC exhibits similar trends in observations with respect to both the evaluation metrics.

In Figure 7, 8 and 9. (seen best in colour), the Pearson's and Kendall's correlations are presented as blue and red bars, respectively. We observe that the performance of the proposed embedding based predictor is in general the best in case of news collections, namely TREC 2 to TREC 8. For web collection topics (TREC

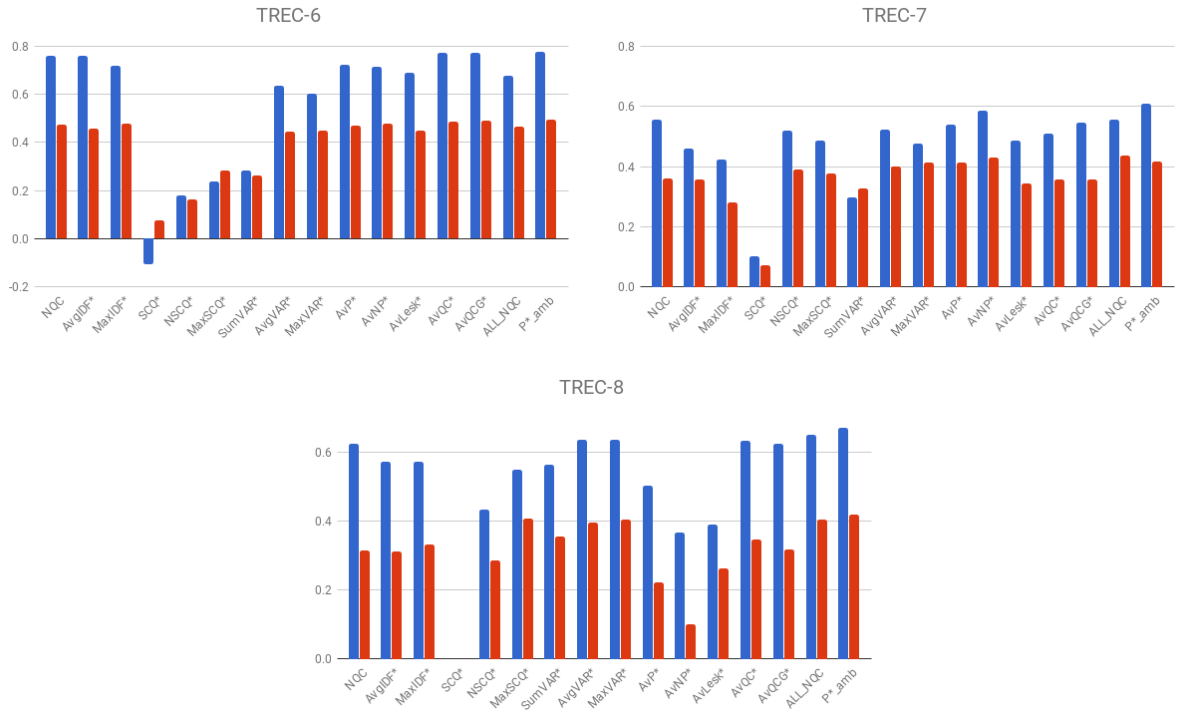

Figure 8: The performance of baseline methods and the proposed methods in terms of performance prediction of BM25 retrieval model on news collection topics TREC 6, 7, and 8. In the plot, Y axis represents the correlation values obtained between the true AP of the retrieval model, and the prediction score. The blue bar and the red bar represent the Pearson's Rho (ρ) and Kendall's Tau (τ) correlation coefficient respectively.

9 and 10), we see that the embedding base approach did not achieve the optimal performance. Specifically, for TREC 9, AvLesk*[14] performs marginally better than $P^*_{clarity}$ in terms of Pearson's correlation (ρ). However, in terms of Kendall's correlation (τ), $P^*_{clarity}$ achieves the best performance. For the other topic set of web collection, i.e. TREC 10, MaxVar [12] attains the best correlations.

It is worth mentioning that the performance of the other predictors are somewhat inconsistent in the sense that they perform satisfactorily for some topic sets but fail to achieve satisfactory effectiveness on others. For example, AvNP* yields the second best Pearson's correlation for TREC 7 but performs relatively poorly on other topic sets. Similar observations can be made for MaxVar*; further, its performance varies considerably for TREC 9 and TREC 10 despite the underlying document collection being the same. Our proposed word embedding based approach, on the other hand, almost always considerably outperforms the baseline methods and that too relatively more consistently than the other ones.

7. Conclusions and Future work

In this paper, we described a novel word embedding based query performance predictor that estimates the ambiguity of a query. Specifically, the query difficulty is quantified by the number of different senses each query term is associated with. The key hypothesis behind the word embedding based predictor is that the word vectors in the local neighbourhood of the query word vectors for an ambiguous query are likely to contain terms associated with different senses. Consequently, identifying the number of senses and the semantic similarity of the query terms with the most prevalent sense is likely to be useful as a predictor function.

Being a pre-retrieval predictor in nature, the embedding based predictor does not utilize any post-retrieval information, such as the similarity score distribution of the top-ranked documents. To assimilate

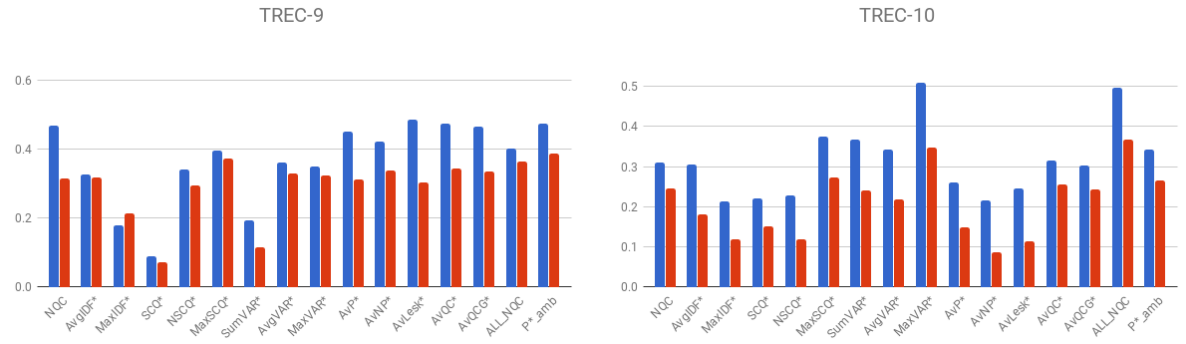

Figure 9: The performance of baseline methods and the proposed methods in terms of performance prediction of BM25 retrieval model on WT10G web collection topics. In the plot, Y axis represents the correlation values obtained between the true AP of the retrieval model, and the prediction score. The blue bar and the red bar represent the Pearson's Rho (ρ) and Kendall's Tau (τ) correlation coefficient respectively.

the post-retrieval knowledge, we further propose a hybrid framework to combine the embedding based predictor with a post-retrieval predictor. In particular, for our experiments with hybrid QPP approaches, we considered NQC, which is a relatively simple to implement and effective post-retrieval performance predictor. Experiments conducted on a number of benchmark TREC datasets demonstrate that the hybrid of the word vector based approach with NQC almost always outperforms combinations of other pre-retrieval predictors with NQC.

In future, we would like to explore the performance of the proposed prediction approach in combination with other post-retrieval methods. In its current form, our proposed predictor uses embeddings to approximate ambiguity of the query ($d(Q, Q)$ in Figure 1). We would like to extend this to use the embeddings of words in the retrieved documents to predict their performance on the basis of the ambiguity of the retrieved documents ($d(R, R)$ in Figure 1). As a generalized predictor, we would also like to utilize the embedded vectors of the top retrieved documents in combination with vectors of neighbouring terms of the query.

References

[1] D. Carmel, E. Yom-Tov, A. Darlow, D. Pelleg, What makes a query difficult?, in: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '06, ACM, New York, NY, USA, 2006, pp. 390–397. doi:10.1145/1148170.1148238.

[2] C. Hauff, D. Hiemstra, F. de Jong, A survey of pre-retrieval query performance predictors, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, ACM, New York, NY, USA, 2008, pp. 1419–1420. doi:10.1145/1458082.1458311.

[3] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. J. Miller, Introduction to wordnet: An on-line lexical database, International journal of lexicography 3 (4) (1990) 235–244.

[4] J. He, M. Larson, M. de Rijke, Using coherence-based measures to predict query difficulty, in: Advances in Information Retrieval, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 689–694.

[5] C. Hauff, Predicting the effectiveness of queries and retrieval systems, SIGIR Forum 44 (1) (2010) 88–88. doi:10.1145/1842890.1842906.

[6] D. Carmel, E. Yom-Tov, Estimating the Query Difficulty for Information Retrieval, Synthesis Lectures on Information Concepts, Retrieval, and Services, Morgan & Claypool Publishers, 2010. doi:10.2200/S00235ED1V01Y201004ICR015.

[7] S. Cronen-Townsend, Y. Zhou, W. B. Croft, Predicting query performance, in: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '02, ACM, New York, NY, USA, 2002, pp. 299–306. doi:10.1145/564376.564429.

[8] Y. Zhou, W. B. Croft, Query performance prediction in web search environments, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07, ACM, New York, NY, USA, 2007, pp. 543–550. doi:10.1145/1277741.1277835.

[9] A. Shtok, O. Kurland, D. Carmel, F. Raiber, G. Markovits, Predicting query performance by query-drift estimation, ACM Trans. Inf. Syst. 30 (2) (2012) 1–:35.

[10] F. Scholer, H. E. Williams, A. Turpin, Query association surrogates for web search: Research articles, JASIST 55 (7) (2004) 637–650. doi:10.1002/asi.v55:7.

[11] B. He, I. Ounis, Inferring query performance using pre-retrieval predictors, in: String Processing and Information Retrieval: 11th International Conference, SPIRE 2004, Padova, Italy, October 5-8, 2004. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 43–54. doi:10.1007/978-3-540-30213-1_5.

[12] Y. Zhao, F. Scholer, Y. Tsegay, Effective pre-retrieval query performance prediction using similarity and variability evidence, in: Advances in Information Retrieval , 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings, 2008, pp. 52–64. doi:10.1007/978-3-540-78646-7_8.

[13] C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to information retrieval, ACM Trans. Inf. Syst. 22 (2) (2004) 179–214. doi:10.1145/984321.984322.

[14] S. Banerjee, T. Pedersen, Extended gloss overlaps as a measure of semantic relatedness, in: Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI'03, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003, pp. 805–810.

[15] R. Rada, H. Mili, E. Bicknell, M. Blettner, Development and application of a metric on semantic nets, in: IEEE Transactions on Systems, Man and Cybernetics, 1989, pp. 17–30.

[16] S. Patwardhan, T. Pedersen, Using wordnet-based context vectors to estimate the semantic relatedness of concepts, in: Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together, Vol. 1501, 2006, pp. 1–8.

[17] M. Sanderson, Word sense disambiguation and information retrieval, in: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, Springer-Verlag New York, Inc., 1994, pp. 142–151.

[18] J. Mothe, L. Tanguy, Linguistic features to predict query difficulty, in: ACM Conference on research and Development in Information Retrieval, SIGIR, Predicting query difficulty-methods and applications workshop, 2005, pp. 7–10.

[19] C. Hauff, L. Azzopardi, D. Hiemstra, The combination and evaluation of query performance prediction methods, in: ECIR, Vol. 5478 of Lecture Notes in Computer Science, Springer, 2009, pp. 301–312.

[20] A. Shtok, O. Kurland, D. Carmel, Query performance prediction using reference lists, ACM Trans. Inf. Syst. 34 (4) (2016) 19:1–19:34. doi:10.1145/2926790.

[21] S. Cronen-Townsend, Y. Zhou, W. B. Croft, A framework for selective query expansion, in: Proceedings of the thirteenth ACM international conference on Information and knowledge management, CIKM '04, ACM, New York, NY, USA, 2004, pp. 236–237.

[22] G. Amati, C. Carpineto, G. Romano, Query difficulty, robustness, and selective application of query expansion, in: Advances in Information Retrieval, 26th European Conference on IR Research, ECIR 2004, Sunderland, UK, April 5-7, 2004, Proceedings, 2004, pp. 127–137. doi:10.1007/978-3-540-24752-4_10.

[23] E. Yom-Tov, S. Fine, D. Carmel, A. Darlow, Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05, ACM, New York, NY, USA, 2005, pp. 512–519. doi:10.1145/1076034.1076121.

[24] V. Vinay, I. J. Cox, N. Milic-Frayling, K. R. Wood, On ranking the effectiveness of searches, in: SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006, 2006, pp. 398–404. doi:10.1145/1148170.1148239.

[25] Y. Zhou, W. B. Croft, Ranking robustness: a novel framework to predict query performance, in: Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006, 2006, pp. 567–574. doi:10.1145/1183614.1183696.

[26] J. A. Aslam, V. Pavlu, Query hardness estimation using jensen-shannon divergence among multiple scoring functions, in: Advances in Information Retrieval, 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007, Proceedings, 2007, pp. 198–209. doi:10.1007/978-3-540-71496-5_20.

[27] F. Diaz, Performance prediction using spatial autocorrelation, in: SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007, 2007, pp. 583–590. doi:10.1145/1277741.1277841.

[28] S. Tomlinson, Robust, web and terabyte retrieval with hummingbird searchserver at TREC 2004, in: Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004, 2004.

[29] Y. Zhou, W. B. Croft, Query performance prediction in web search environments, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07, ACM, New York, NY, USA, 2007, pp. 543–550. doi:10.1145/1277741.1277835.

[30] O. Butman, A. Shtok, O. Kurland, D. Carmel, Query-performance prediction using minimal relevance feedback, in: Proceedings of the 2013 Conference on the Theory of Information Retrieval, ICTIR '13, ACM, New York, NY, USA, 2013, pp. 7:14–7:21. doi:10.1145/2499178.2499201.

[31] H. Roitman, An enhanced approach to query performance prediction using reference lists, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, ACM, New York, NY, USA, 2017, pp. 869–872. doi:10.1145/3077136.3080665.
URL http://doi.acm.org/10.1145/3077136.3080665

[32] H. Roitman, S. Erera, B. Weiner, Robust standard deviation estimation for query performance prediction, in: Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR '17, ACM, New York, NY, USA, 2017, pp. 245–248. doi:10.1145/3121050.3121087.
URL http://doi.acm.org/10.1145/3121050.3121087

[33] A. Shtok, O. Kurland, D. Carmel, Predicting query performance by query-drift estimation, in: Advances in Information Retrieval Theory, Second International Conference on the Theory of Information Retrieval, ICTIR 2009, Cambridge, UK,

September 10-12, 2009, Proceedings, 2009, pp. 305–312. doi:10.1007/978-3-642-04417-5_30.
[34] R. Cummins, J. Jose, C. O'Riordan, Improved query performance prediction using standard deviation, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11, ACM, New York, NY, USA, 2011, pp. 1089–1090. doi:10.1145/2009916.2010063.
[35] D. Pal, M. Mitra, S. Bhattacharya, Using multiple query expansion algorithms to predict query performance, in: Proceedings of the Fourth International Conference of Emerging Applications of Information Technology, EAIT '14, 2014, pp. 361–364. doi:10.1109/EAIT.2014.67.
[36] M. Hasanain, T. Elsayed, Query performance prediction for microblog search, Information Processing & Management 53 (6) (2017) 1320 – 1341. doi:https://doi.org/10.1016/j.ipm.2017.08.002.
URL http://www.sciencedirect.com/science/article/pii/S0306457317300614
[37] F. Raiber, O. Kurland, Query-performance prediction: Setting the expectations straight, in: Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, ACM, New York, NY, USA, 2014, pp. 13–22. doi:10.1145/2600428.2609581.
[38] O. Kurland, A. Shtok, S. Hummel, F. Raiber, D. Carmel, O. Rom, Back to the roots: a probabilistic framework for query-performance prediction, in: 21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012, 2012, pp. 823–832. doi:10.1145/2396761.2396866.
[39] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Curran Associates Inc., USA, 2013, pp. 3111–3119.
[40] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation., in: EMNLP, Vol. 14, 2014, pp. 1532–1543.
[41] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, TACL 5 (2017) 135–146.
[42] B. Salehi, P. Cook, T. Baldwin, A word embedding approach to predicting the compositionality of multiword expressions, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015, pp. 977–983.
[43] M. Del Tredici, N. Bel, A word-embedding-based sense index for regular polysemy representation, in: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, 2015, pp. 70–78.
[44] W. Gharbieh, V. Bhavsar, P. Cook, A word embedding approach to identifying verb-noun idiomatic combinations, in: Proceedings of the 12th Workshop on Multiword Expressions, 2016, pp. 112–118.
[45] A. Joshi, V. Tripathi, K. Patel, P. Bhattacharyya, M. Carman, Are word embedding-based features useful for sarcasm detection?, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1006–1011.
[46] L. Boytsov, D. Novak, Y. Malkov, E. Nyberg, Off the beaten path: Let's replace term-based retrieval with k-nn search, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, ACM, New York, NY, USA, 2016, pp. 1099–1108. doi:10.1145/2983323.2983815.
URL http://doi.acm.org/10.1145/2983323.2983815
[47] E. Nalisnick, B. Mitra, N. Craswell, R. Caruana, Improving document ranking with dual word embeddings, in: Proceedings of the 25th International Conference Companion on World Wide Web, WWW '16 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2016, pp. 83–84. doi:10.1145/2872518.2889361.
URL https://doi.org/10.1145/2872518.2889361
[48] D. Ganguly, D. Roy, M. Mitra, G. J. Jones, Word embedding based generalized language model for information retrieval, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, ACM, New York, NY, USA, 2015, pp. 795–798. doi:10.1145/2766462.2767780.
URL http://doi.acm.org/10.1145/2766462.2767780
[49] I. Vulić, M.-F. Moens, Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, ACM, New York, NY, USA, 2015, pp. 363–372. doi:10.1145/2766462.2767752.
URL http://doi.acm.org/10.1145/2766462.2767752
[50] G. Zuccon, B. Koopman, P. Bruza, L. Azzopardi, Integrating and evaluating neural word embeddings in information retrieval, in: Proceedings of the 20th Australasian Document Computing Symposium, ADCS '15, ACM, New York, NY, USA, 2015, pp. 12:1–12:8. doi:10.1145/2838931.2838936.
URL http://doi.acm.org/10.1145/2838931.2838936
[51] F. C. Fernndez-Reyes, J. Hermosillo-Valadez, M. M. y Gmez, A prospect-guided global query expansion strategy using word embeddings, Information Processing & Management 54 (1) (2018) 1 – 13. doi:https://doi.org/10.1016/j.ipm.2017.09.001.
URL http://www.sciencedirect.com/science/article/pii/S0306457317301140
[52] D. Roy, D. Ganguly, M. Mitra, G. J. Jones, Word vector compositionality based relevance feedback using kernel density estimation, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, ACM, New York, NY, USA, 2016, pp. 1281–1290. doi:10.1145/2983323.2983750.
[53] D. Roy, D. Paul, M. Mitra, U. Garain, Using word embeddings for automatic query expansion, in: Proc. of NeuIR Workshop, collocated with SIGIR, 2016.
[54] H. Zamani, W. B. Croft, Estimating embedding vectors for queries, in: Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR '16, ACM, New York, NY, USA, 2016, pp. 123–132. doi:

10.1145/2970398.2970403.
[55] R. Yan, Y. Song, H. Wu, Learning to respond with deep neural networks for retrieval-based human-computer conversation system, in: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '16, ACM, New York, NY, USA, 2016, pp. 55–64. doi:10.1145/2911451.2911542.
URL http://doi.acm.org/10.1145/2911451.2911542
[56] G. Zhou, T. He, J. Zhao, P. Hu, Learning continuous word embedding with metadata for question retrieval in community question answering, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, 2015, pp. 250–259.
[57] Q. Zhang, J. Kang, J. Qian, X. Huang, Continuous word embeddings for detecting local text reuses at the semantic level, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '14, ACM, New York, NY, USA, 2014, pp. 797–806. doi:10.1145/2600428.2609597.
URL http://doi.acm.org/10.1145/2600428.2609597
[58] K. D. Onal, Y. Zhang, I. S. Altingovde, M. M. Rahman, P. Karagoz, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, A. Angert, E. Banner, V. Khetan, T. McDonnell, A. T. Nguyen, D. Xu, B. C. Wallace, M. de Rijke, M. Lease, Neural information retrieval: at the end of the early years, Information Retrieval Journaldoi:10.1007/s10791-017-9321-y.
URL https://doi.org/10.1007/s10791-017-9321-y
[59] J. Li, D. Jurafsky, Do multi-sense embeddings improve natural language understanding?, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, 2015, pp. 1722–1732.
URL http://aclweb.org/anthology/D/D15/D15-1200.pdf
[60] B. Athiwaratkun, A. Wilson, Multimodal word distributions, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2017, pp. 1645–1656. doi:10.18653/v1/P17-1151.
URL http://www.aclweb.org/anthology/P17-1151
[61] A. Trask, P. Michalak, J. Liu, sense2vec - A fast and accurate method for word sense disambiguation in neural word embeddings, CoRR abs/1511.06388. arXiv:1511.06388.
URL http://arxiv.org/abs/1511.06388
[62] A. Cuba Gyllensten, M. Sahlgren, Navigating the semantic horizon using relative neighborhood graphs, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2015, pp. 2451–2460. doi:10.18653/v1/D15-1292.
URL http://www.aclweb.org/anthology/D15-1292
[63] H. Zamani, W. B. Croft, J. S. Culpepper, Neural query performance prediction using weak supervision from multiple signals, in: Proc. of SIGIR '18, 2018, pp. 105–114.
[64] F. Diaz, B. Mitra, N. Craswell, Query expansion with locally-trained word embeddings, in: Proc. of ACL (1), The Association for Computer Linguistics, 2016.
[65] N. Rekabsaz, M. Lupu, A. Hanbury, Exploration of a threshold for similarity based on uncertainty in word embedding, in: European Conference on Information Retrieval, Springer, 2017, pp. 396–409.
[66] J. He, M. Larson, M. de Rijke, Using coherence-based measures to predict query difficulty, in: Advances in Information Retrieval , 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings, 2008, pp. 689–694. doi:10.1007/978-3-540-78646-7_80.
[67] S. Onn, I. Weissman, Generating uniform random vectors over a simplex with implications to the volume of a certain polytope and to multivariate extremes, Annals OR 189 (1) (2011) 331–342.
[68] M. G. Kendall, Rank Correlation Methods, New York: Hafner Publishing Co., New York, 1955.
[69] D. Hiemstra, Using language models for information retrieval, Ph.D. thesis, Center of Telematics and Information Technology, AE Enschede (2000).
[70] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Found. Trends Inf. Retr. 3 (4) (2009) 333–389. doi:10.1561/1500000019.
[71] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.