Evaluating the Effectiveness of Query-Document Clustering Using the QDSM Measure

Guti´errez-Soto, C; Palomino, M; Curiel, A; Cerda, HER; Rain, FB

Published in:
Advances in Science, Technology and Engineering Systems Journal

DOI:
10.25046/aj0506105

Publication date:
2020

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 5, No. 2, XX-YY (2020)
www.astesj.com

ASTES Journal
ISSN: 2415-6698

Evaluating the Effectiveness of Query-Document Clustering Using the QDSM Measure

Claudio Gutiérrez-Soto[*,1], Marco Palomino[2], Arturo Curiel[3], Héctor Riquelme Cerda[1], Fernando Bejar Rain[1]

[1]*Universidad del Bío-Bío, Department of Information Systems, Casilla 5-C, Chile*

[2]*University of Plymouth, School of Engineering, Computing and Mathematics, PL4 8AA, England*

[3]*CONACYT-Universidad Veracruzana, Facultad de Estadística e Informática, 91020, México*

ARTICLE INFO

Article history:
Received:
Accepted:
Online:

Keywords:
Web search engines;
Query-sensitive similarity measure (QSSM);
Query logs;
Ward's method;
Average link

ABSTRACT

It is well documented that the average length of the queries submitted to Web search engines is rather short, which negatively impacts the engines' performance, as measured by the precision metric. It is also well known that ambiguous keywords in a query make it hard to identify what exactly search engine users are looking for. One way to tackle this challenge is to consider the context in which the query is submitted, making use of query-sensitive similarity measures (QSSM). In this paper, a particular QSSM known as the query-document similarity measure (QDSM) is evaluated, QDSM is designed to determine the similarity between two queries based on their terms and their ranked lists of relevant documents. To this extent, F-measure and the nearest neighbor (NN) have been employed to assess this approach over a collection of AOL query logs. Final results reveal that both the Average Link Algorithm and Ward's method present better results using QDSM than cosine similarity.

1 Introduction

This paper is an extension of work originally presented in the International Conference of the Chilean Computer Science Society (SCCC) [1]. Nowadays, exist a large amount of information available on the Web; additionally, web search engines (WSE) daily index thousands of pages, by which finding relevant and timely information over this growth without constraints becomes quite a challenge. According to [2], this unrestricted growth has not been accompanied by corresponding technical advances in approaches to extract relevant information. Unsuccessful searches are common in WSEs and can be given by several reasons among which we can mention the following. First, the lengths of submitted queries by users are mostly short (e.g., the average size of a web search is 2.4 words [3]). Owing to queries are conformed by a few keywords, it is complicated to determine the specific topic in which the query is inserted, therefore, a result with a few keywords in the query can be formed by unrelated topics. Furthermore, users ignore how to formulate a correct query [4]. The problem is made more complex when the user does not have a specific idea about which should be the result that he/she is looking for (i.e., which turns in relevant information for him/her). In light of the foregoing, it is not easy for WSEs to interpret the meaning of what users are looking for. One way to tackle this issue, is to consider the context in which the queries are submitted at WSEs [5][6]. To capture the context, WSEs should consider what queries are related among them (i.e., determining if the queries are similar) and how their results have been beneficial for users. A way to determine the relationship between similar queries and relevant documents for these queries is given by the cluster hypothesis, which establishes that all documents considered as relevant for a query are similar to each other (i.e., similar documents can be relevant for the same query)[7]. Accordingly, it can be assumed that relevant documents for a query q, are relevant for a query q', such as q' is similar to q. Thus, having methods that provide the similarity among documents and queries can bring a better characterization about the meaning of a new query, and as a consequence, it entails more effective results.

Whether the WSE is able to establish how similar is a new query regarding queries recently submitted, then the search engine should provide documents, which were relevant in previous searches. Hence, the recent past queries along with the relevant documents provide a context, in which is feasible to improve the answers to

[*]Corresponding Author Name, Department of Information Systems, Universidad del Bío-Bío, Casilla 5-C, Email: cogutier@ubiobio.cl

new searches [8]. Nevertheless, measuring how similar are two queries using their documents is not easy, suitable metrics that allow representing the context are needed.

Aiming to capture the context in which queries are submitted, Tombros and Van Rijsbergen [9] present a pioneering approach introducing the measuring called query-sensitive similarity measure (QSSM). This measure establishes the following; two documents are more similar than others, whether both are more similar regarding a given query. Following this argumentation line, QSSMs can be used as a metric to measure the similarity between two queries considering the context. As such, a WSE using additional information can improve its effectiveness to answer a new query. To achieve this goal, queries alongside their documents should be stored in clusters. Currently, few approaches store the queries along with their documents [10][11][12][13] [14] (these approaches recommend to the user a list of similar queries, which are related to the submitted query by the user). However, the approaches mentioned previously are not directly related to the approach presented in this paper.

1.1 Contribution

The main contribution of this paper is the effectiveness evaluation of QDSM. Roughly speaking, effectiveness is related to the quality of recovered documents. Better effectiveness occurs when more relevant documents are retrieved (from a total of N documents retrieved, there are more relevant documents than non-relevant documents). By contrast, worse effectiveness occurs when more non-relevant documents are retrieved. With this in mind, grouping similar queries alongside their relevant documents (in clusters) should directly affect the effectiveness. Therefore, improving clusters' effectiveness implies having more relevant documents by clusters, which aligns with the cluster hypothesis.

To evaluate QDSM effectiveness, the F-measure alongside the nearest-neighbor (NN) cluster hypothesis tests were used. Both tests were applied over the following algorithms; Single Link, Complete Link, Average Link, Bisection K-means, and Ward's method. Three relevance models were simulated with the aim to determine which documents are relevant for a specific query. Besides, a wide range of experiments was carried out over five sets of queries. Final results are presented as QDSM improvement percents regarding cosine measure (S_c) (or cosine similarity), which were contrasted with the values obtained applying the Student Paired T-test (two samples).

The remainder of this paper is organized as follows: In Section 2, a detailed review of articles related to the problem of capturing the context using clustering is presented. In Section 3, the methodological description, is exposed. Section 4 presents the experimental environment. Section 5 displays the empirical results, which are then discussed in Section 6. Finally, Section 7 gives some perspectives of future work along with the conclusions.

2 Related Work

There have been many works that deal with the use of clustering in Information Retrieval (IR). Clustering in IR has been employed to improve the effectiveness (i.e., quality of clusters). Overall, clustering-based approaches that intend to capture the context in which the queries are submitted can be classified into two categories, considering the underlying repositories (these are also known as collections or datasets). The first category involves using traditional IR datasets. Some of them use QSSM similarities such as [15], [16], [17]; meantime, the second category uses log-file data from search engines.

To improve effectiveness in the retrieval process, an approach relies on hierarchic query-specific clustering is presented in [15]. In pursuing this goal, a wide range of experiments was performed. According to the authors, given a specific query, the hierarchy should be adapted to increase the likelihood to situate relevant documents to the query in nearby clusters. Two characteristics stand out in this research. First, an analysis of optimal clusters variation considering the number of top-ranked documents allowing better effectiveness is exposed. Finally, a comparison between their results and inverted file search (IFS) is provided. Five traditional IR collections alongside four hierarchic agglomerative methods were employed in all experiments. Final results indicate that query-specific clustering outweighs static clustering in each of the experiments. On the other side, a framework based on probabilistic co-relevance, which gives a query-sensitive similarity, is presented in [17]. The similarity between two documents corresponds to the ratio between the co-relevance probability and a specific query. Two cases were considered to identify the co-relevance. First, the document's relevance is independent of the rest of the documents. Second, the document's relevance is dependent on the rest. Several experimental scenarios were studied using the nearest neighbor test on TREC collections. The final results reveal that the framework outperforms term-based similarity.

The approaches mentioned above expand the users' judgments grounded on the following assumption. All terms included in a relevant document for a specific query are relevant too. Consequently, it is assumed that all documents that include some of these terms are also relevant. Besides, these approaches do not deal with the similarity among queries, with the exception of [18][16]. In [18], a method called Scatter/Gather, which explores clusters based on documents, is proposed. The method returns a ranked title's list for the organization and viewing of retrieval results. Scatter/Gather is used as a tool for retrieval of browsing results, which presents summaries to users. Towards that goal, documents are joined in similar topics. A fractional algorithm provides k clusters on TREC/Tipster dataset. As a result of experimentation, the authors assert that their method gives tailored clusters according to the query's characteristics. In such a way, their results corroborate the cluster hypothesis since relevant documents are more similar to each other than non-relevant documents. In [16], the authors introduce the Weighted Borda (WBorda) model, which determines the co-relevance of a document using different similarities' types. To this end, a Support Vector Machine (SVM) was trained to get the

estimated co-relevance, fusing the induced rankings using several functions. Each function considers the similarity between documents and the query. Several similarity measures were considered in experiments such as cosine BM25, M1, and M3. The final results in tasks such as nearest-neighbor clustering, cluster-based, and graph-based document retrieval indicate that WBorda provides better results than several proposed co-relevance models.

On the other hand, approaches such as [19][10], and [21] belong to the second category. Users' log-based, an approach of query clustering is proposed in [19]. Towards that end, the documents previously read by users are employed to construct cross-references among documents and queries. According to the researchers exist a strong relationship between the selected documents and queries. This approach underlies two fundamental aspects: First, two queries are similar if users clicked on the same documents; Second, if a set of documents was selected for the same queries, then the documents' terms are related to the queries' terms. The empirical results were obtained using the DBSCAN algorithm and the Encarta encyclopedia dataset. The final results show that many similar queries are gathered in the same clusters utilizing this approach. A query-clustering classification, which compares various query similarity measures, is presented in [10]. Three groups: content-based approaches, feedback-based approaches, and results-based approaches are suggested in this classification. In content-based approaches, the common terms of queries are used to describe query clusters. Similarity functions such as Jaccard, Cosine, and Dice were employed to build the clusters. In that regard, the authors claim that this method is not convenient for search engines due to many queries have few terms. On the other side, in feedback-based approaches, the similarity measure is grounded users selections over search results; therefore, two queries are similar whether they encourage the selection of similar documents. In turn, results-based approaches evaluate the similarity between queries through the overlap of returned documents. In this case, the researchers point out that this approach's principal drawback corresponds to high processing times. Notable results are obtained using the three approaches in parallel. In [21], a WSE provides a user with a list of similar queries regarding the user's submitted query. Semantically similar queries give support to the clustering process. Clusters are formed, taking into account the historical preferences of registered users in the WSE. To build the clusters, term-weight vector representation of queries considering the clicked URLs was employed. The method exhibits two benefits, (1) it discovers the related queries, and (2) sorts the queries rely on a relevance criterion. It is important to mention that the queries are sorted using the following criteria: (a) the similarity between the clusters' queries and the new query and (b) the support, which is related to how much the query answers capture the user interest. The experiments were conducted using the combination of (a) and (b). The results display improvements on average precision.

In summary, the first category is based on traditional IR datasets. A traditional IR dataset is formed by three sets, a set of documents (D), a group of queries (Q), and a set of users' judgments (JU). The user's judgments contain the relevant documents for a query in Q. Note that all works mentioned in this category include new relevant documents (if some document has some relevant term, then it is considered relevant), which are not part of the original JU. On the other hand, it should be noted that there are no JU in the second category (log-files from search engines). Consequently, subject matter experts evaluate the pertinence of a document given a query. Note that all approaches mentioned in this related work modified some documents' relevance, which directly impacts effectiveness. Contrary to these approaches, in this paper, three types of users' judgments are simulated without altering the documents' original-relevance.

The overall procedure and a discussion about the results are presented in the following sections.

3 Methodology

The methodology overview is as follows. Initially, a user submits a query to the WSE. Thereafter, the WSE returns the documents as a result of the query. These documents are ranked from the most similar to the least similar regarding the query. Once this is done, the documents are stored along with the query in clusters inside the WSE. Aiming to form the clusters considering documents and queries, QDSM is used. In this way, when a user submits a new query, it is contrasted with the past queries (these are the queries previously stored) in the query-document clusters. Accordingly, an effectiveness improvement should occur due to the clusters closest to the new query containing relevant documents for the new query.

Document relevances become a crucial factor in enhancing effectiveness. In a traditional IR dataset, document relevances are given by subject matter experts, who determine what documents are relevant given a query. These documents are reflected in the users' judgments. On the other hand, the relevance of documents in a WSE is given by ranking functions. Overall, ranking functions attempt to capture the relevance through users' clicks on documents, which are expressed in the ItemRanks. In this manner, a retrieved document (i.e., URL or web page) with an ItemRank high could be considered as relevant.

It is essential to keep in mind that most approach clustering-based extend or use subject matter experts to give relevance to the documents, due to none of these datasets have been designed to work with similar queries (past queries are part of clusters). According to [22], a good way to tackle this problem is by using simulation. In this paper, document relevances have been simulated. Two notable advantages are obtained with the simulation use. First, it is neither necessary to use subject matter experts nor extend the users judgments. Second, several models of relevance (a model can be seen as a ranking function) can be used; for instance, given a query, a document can be relevant or non-relevant depending on the model. In this paper, this is given by a relevance function, which determines the relevance of a document considering both its corresponding ItemRank and relevance probability.

Aiming to shed light on how QDSM is evaluated using a relevance function, suppose the following example. Assume that three

documents (d_5, d_{10}, and d_{12}) have been recovered for a query q, such as d_5 is the most similar document concerning the query. The respective ItemRank for each document is 20, 25, and 30. Likewise, the probabilities of relevance according to their respective ItemRanks are 90%, 40% 70%. Additionally, suppose a relevance function that only considers the last recovered document (in this case, d_{12}). As d_{12} has an ItemRank of 30, the probability of being relevant is 70%. To simulate the relevance probability of d_{12}, a binary array of 100 elements is used. Initially, this array is instantiated with 0 values; subsequently, 70 random positions with value 1 are assigned in the array using Uniform Distribution. In order to give the relevance to d_{12}, an array position is selected using Uniform Distribution; thus, if this value is 1, then d_{12} is relevant; in another case, d_{12} is non-relevant (Note that d_5 and d_{10} are non-relevant).
Following the same example, suppose a relevance function that assigns the relevance individually, then the same procedure to give relevance is performed for each document (d_5, d_{10}, and d_{12}). Thus, a possible result could be that d_5 and d_{12} being relevant, meantime d_{10} could be non-relevant. Finally, suppose a relevance function that provides the average relevance, then the average of ItemRanks is obtained, and its relevance probability is used to give relevance to the three documents.

Note that different relevance functions could provide different results on QDSM, due to QDSM considers the relevant documents as part of its metric.

3.1 The Query-Document Similarity Measure

The Query-Document Similarity Measure (QDSM) is a Query-Sensitive Similarity Measure (QSSM), which has as a fundamental purpose to capture the semantic similarity between queries, taking into consideration terms that belong to the queries as well as the position in which appear the relevant documents in both lists. Indirectly, the terms associated with the relevant documents should contribute to providing context. Specifically, each list of documents is presented in descending order according to the similarity of documents regarding the query. From the semantic point of view, two queries are closer if they share more relevant documents in their lists. This can be appreciated by observing the number of relevant documents that form the intersection between the two lists. Therefore, while more relevant documents make up the intersection, the more similar the queries will be. Thus, this paper's primary assumption is that using similar queries alongside their relevant documents should provide clusters with better effectiveness than S_c, since additional information can be captured from documents, including the queries (i.e., information is not complete in each query individually). Specifically, this additional information is given by the union of queries and documents' terms but does not belong to the intersection among them. Using this rationale, QDSM is in line with the cluster hypothesis, which claims that relevant documents for a particular query tend to be close, whereby these relevant documents should tend to be in the same cluster for a specific query.

QDSM takes advantage from the place in which relevant documents appear on the list. As reported by [23], the most similar documents concerning the query tend to appear at the beginning of the list. On this basis, the order in which relevant documents appear in both lists gives information about the context (particularly the terms of relevant documents). QDSM deals with the order of relevant documents through the use of the Longest Common Subsequence (LCS) algorithm. LCS allows acquiring the relative similarity keeping the order in which simultaneously appears a relevant document in both queries. By doing so, the context capturing in which the queries are submitted is possible.

This measure is convenient in two situations:

- When terms of a query are few, as is currently happening in the WSEs.
- In a dynamic environment, where the documents' relevance could change (i.e., the position of a document in the list could change as well as its ItemRank), non-relevant documents could become relevant documents.

Accordingly, the queries are either short length (i.e., few keywords in the query) or ambiguous. Nevertheless, these can be enriched with more information associated with their relevant-documents retrieved.

Aiming to give formality, some definitions are detailed below.

Definition 1. Let D be a set of documents, such as every document in D, is formed by a set of terms (i.e., words contained in d). D is stored in a WSE W. Besides, let q be a single query, such as $q \in Q$, where Q is a set of queries interpretable by W.

Definition 2. The cosine measure between q and d_i, is defined as:

$$S_c(q, d_i) = \frac{\sum_{j=1}^{m} t_{qj} \cdot t_{ij}}{\sqrt{\sum_{j=1}^{m} t_{qj}^2 \cdot \sum_{j=1}^{m} t_{ij}^2}}$$

shuch as $d_i \in D \wedge q \in Q$, t_q are the query'terms and t_i are the document's terms.

Definition 3. Let $|d_i|$ be the number of terms in a document d_i, such as $d_i \in D$. Note that this definition can be applied to obtain the number of terms for a query q.

Lemma 1. $S_c(q, d_i) \neq 0 \iff (q \cap d_i) \neq \emptyset$
Proof by contradiction. Suppose $(q \cap d_i) = \emptyset$. Then, there are no terms in common between q and d_i. Hence, applying the *Definition 2* implies $S_c(q, d_i) = 0$. Likewise, if $S_c(q, d_i) = 0$, then $(q \cap d_i) = \emptyset$. □

Definition 4. Let $\mathcal{L}(q) = \{d_i | S_c(q, d_i) \neq 0 \ \forall d_i \in D\}$ be the set of documents whose similarity with $q \in Q$ is different to 0.

Definition 5. Let $\mathcal{L}_N(q)$ be a list of N retrieved documents from W, such as $\mathcal{L}_N(q)$ is ranked by decreasing order (i.e., they are ordered from highest to lowest according to S_c).

Definition 6. Let $F(ItemRank(d_i), M)$ be a binary function, which provides relevance to a document d_i, such as 1 is relevant and 0 is

non-relevant. The function is defined as follows:

$$F(ItemRank(d_i), M) = \begin{cases} 1: & Pr(1) = \mathcal{M}(M, ItemRank(d_i)) \\ 0: & Pr(0) = 1 - Pr(1) \end{cases}$$

where M corresponds to the relevance model (i.e., PartialRel, AvRel or LastRel). *ItemRank* is a function that provides the rank for the document d_i, and $\mathcal{M}$ is the function that gives the probability considering M and *ItemRank*. A binary array formed by 100 elements is used to represent the probability in $\mathcal{M}$. The Uniform Distribution is employed to instantiate the values (the percentage is represented with values 1 in the array) and determine the relevance (if the array's selected position contains a 1, then the document is relevant).

Definition 7. Let $\mathcal{L}_{N,R}(q)$ be a list of retrieved documents along with their relevances, then:
$\mathcal{L}_{N,R}(q) = \{(d_i, F(ItemRank(d_i), M))|d_i \in \mathcal{L}_N(q)\}$.

Definition 8. Given two queries q and q' such as both queries are in Q, and their corresponding lists of documents $\mathcal{L}_{N,R}(q)$ and $\mathcal{L}_{N',R}(q')$. Then QDSM is defined as follow:

$QDSM(q, q') = \frac{S_c(q,q') + LCS(\mathcal{L}_{N,R}(q), \mathcal{L}_{N',R}(q'))}{1 + max(\|\mathcal{L}_{N,R}(q)\|, \|\mathcal{L}_{N',R}(q')\|)}$
where:

- $S_c(q, q')$ corresponds to the cosine measure between the queries q and q'.
- $LCS(\mathcal{L}_{N,R}(q), \mathcal{L}_{N',R}(q'))$ is the LCS algorithm applied over $\mathcal{L}_{N,R}(q)$ and $\mathcal{L}_{N',R}(q')$ [24].
- *max* gives the greatest number of relevant documents between the lists $\mathcal{L}_{N,R}(q)$ and $\mathcal{L}_{N',R}(q')$.

Theorem 1. $QDSM(q, q') = 0 \iff S_c(q, q') = 0 \wedge LCS(\mathcal{L}_{N,R}(q), \mathcal{L}_{N',R}(q')) = 0$.
Proof by contradiction.

- Suppose $S_c(q, q') \neq 0 \vee LCS(\mathcal{L}_{N,R}(q), \mathcal{L}_{N',R}(q')) \neq 0$. If $S_c(q, q') \neq 0$ is enough for $QDSM(q, q') \neq 0$ (by *Definition 8*). Likewise, as $LCS(\mathcal{L}_{N,R}(q), \mathcal{L}_{N',R}(q')) \neq 0$ then $QDSM(q, q') \neq 0$ (by *Definition 8*).
- Finally, as $QDSM(q, q') \neq 0$ then either $S_c(q, q') \neq 0 \vee LCS(\mathcal{L}_{N,R}(q), \mathcal{L}_{N',R}(q')) \neq 0$ (by *Definition 8*). □

Lemma 2.
$LCS(\mathcal{L}_{N,R}(q), \mathcal{L}_{N',R}(q')) \neq 0 \iff \exists\, t \in (d \cap q \cap q')$
Proof.

- $LCS(\mathcal{L}_{N,R}(q), \mathcal{L}_{N',R}(q')) \neq 0$, then $\exists\, d \in (\mathcal{L}_{N,R}(q) \cap \mathcal{L}_{N',R}(q'))$ (by the LCS invariant [24]). Thus, $d \in \mathcal{L}_{N,R}(q) \wedge d \in \mathcal{L}_{N',R}(q')$. Since $d \in \mathcal{L}(q) \wedge d \in \mathcal{L}(q')$, then $S_c(d, q) \neq 0 \wedge S_c(d, q') \neq 0$ (by *Definition 4*). Thus, $(d \cap q) \neq \emptyset \wedge (d \cap q') \neq \emptyset$ (by *Lemma 1.*). Hence, $(d \cap q \cap q') \neq \emptyset$. Therefore, $\exists\, t \in (d \cap q \cap q')$.
- Finally, as $\exists\, t \in (d \cap q \cap q')$ then $S_c(q, q') \neq \emptyset \wedge S_c(d, q) \neq \emptyset \wedge S_c(d, q') \neq \emptyset$ (by *Lemma 1.*). Thus $d \in \mathcal{L}_{N,R}(q) \wedge d \in \mathcal{L}_{N',R}(q')$ (by *Definition 7.*). Therefore, $LCS(\mathcal{L}_{N,R}(q), \mathcal{L}_{N',R}(q')) \neq 0$ (by *Theorem 1*). □

Lemma 2 asserts that exists at least a common document in both lists, and therefore at least there is one term in common among queries and the document.

Lemma 3.
$|d| \neq |q| \implies \exists t \in (d \triangle q)$
Proof. Suppose $|d| > |q|$, which implies that d has at least one term more than q. On the other hand, can ocurr that $|q| > |d|$, then q has at least one term more than d. Thus, $\exists t \in ((d - q) \cup (q - d)$. Therefore, $\exists t \in (d \triangle q)$. □

Lemma 3 points out that if different numbers of terms form the document and the query, then at least there is one term that does not belong to the intersection between them. Note that d and q are in $\mathcal{L}_{N,R}(q)$.

Theorem 2.
$QDSM(q, q') \neq 0 \wedge |d| \neq |q| \neq |q'| \iff \exists\, t \in ((d \triangle q) \triangle q')$; such as d is in both $\mathcal{L}_{N,R}(q) \wedge \mathcal{L}_{N',R}(q')$.
Proof.

- $QDSM(q, q') \neq 0 \wedge |d| \neq |q| \neq |q'|$, then $LCS(\mathcal{L}_{N,R}(q), \mathcal{L}_{N',R}(q')) \neq 0 \wedge S_c(q, q') \neq \emptyset \wedge |d| \neq |q| \neq |q'|$ (by *Definition 8*). As the first part, consider $LCS(\mathcal{L}_{N,R}(q), \mathcal{L}_{N',R}(q')) \neq 0 \wedge |d| \neq |q| \neq |q'|$ then, $\exists\, t' \in (d \cap q \cap q') \wedge |d| \neq |q| \neq |q'|$ (by *Lemma 2*). Thus $\exists\, t' \in (d \cap q \cap q') \wedge ((|d| \neq |q|) \neq |q'|)$. Subsequenlty, $\exists\, t' \in (d \cap q \cap q') \wedge \exists t \in ((d \triangle q) \neq q')$. (by *Lemma 3.*). Hence, $\exists\, t \in ((d \triangle q) \triangle q')$. The another case is $S_c(q, q') \neq \emptyset$, then $\exists\, t \in (q \triangle q')$, considering d in the hypothesis, $(q \triangle q') \subset ((d \triangle q) \triangle q')$. Therefore $\exists\, t \in ((d \triangle q) \triangle q')$.
- Finally and without loss of generality, if $\exists\, t \in ((d \triangle q) \triangle q')$ then $\exists\, t \in ((d - q) \cup (q - d)) \vee ((d - q') \cup (q' - d)) \vee ((q - q') \cup (q' - q)) \wedge ((S_c(q, q') \neq \emptyset \wedge \exists\, t \in (d \cap q \cap q'))$ (by *Lemma 1. and Lemma 2.)* then $QDSM(q, q') \neq 0 \wedge |d| \neq |q| \neq |q'|$. □

Definition 9. Let RQ be a set of queries, along with their retrieved documents and their corresponding relevances, then RQ is defined as follow:

$$RQ = \bigcup_{i=1}^{|Q|} \mathcal{L}_{N,R}(q_i)$$

Definition 10. Let $\widehat{D}$ be a benchmark query set, which is formed by Z subsets of queries, such as:

$$\widehat{D} = \bigcup_{i=1}^{Z} RQ$$

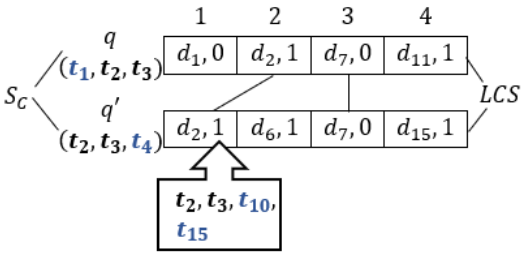

Figure 1: QDSM measure

Theorem 2 ensures that the context in which the queries are submitted in the WSE can be captured by the complementary terms to both queries and their relevant documents (i.e., the symmetric difference of sets d, q and q' $((d \triangle q) \triangle q')$). An example of the Theorem 2 essence and how QDSM is computed, is displayed in Figure 1. In Figure 1, the common terms for both queries are presented in bold (i.e., these are t_2 and t_3), note that both terms are common in d_2, which is a relevant document (i.e., all document d_i, 1 are relevant meanwhile d_i, 0 are non-relevant). LCS is applied over both lists of retrieved documents considering only the relevant documents in the lists (i.e., even though d_7 is in both lists, only d_2 is considered). Finally, all terms that give context are in blue color (i.e., t_1, t_4, t_{10} and t_{15}).

It is worth noting that QDSM takes the value 1 (see *Definition 8*) when q and q' are the same queries, and all retrieved documents are relevant. Specifically, this latter can be itemized in two parts. In the first part, $S_c(q, q')$ provides 1 because q and q' are the same. In the second part, both results lists are equal, and all retrieved documents are relevant; therefore, both lists hold the relevant documents in the same positions.

In summary, QDSM provides a metric that captures the semantic relationship between two queries (context), considering the relevant documents' order in both lists. A wide range of experiments was conducted in order to compare the effectiveness between S_c and QDSM. The experimental setup is displayed in the following section.

4 Experimental Environment

A benchmark query set extracted from the well-known dataset of query log "AOL Query Logs Dataset (AOL) [25]", was used to carry out the experiments. This collection has more than 20 million web query logs stored, submitted by around 650 thousand users in more than 36 thousand lines of data. These queries were stored at an interval of three months in the year 2006. Broadly speaking, queries in AOL are depicted as rows in the database files, which contains five columns with the following fields:
{*AnonID*, *Query*, *QueryTime*, *ItemRank*, *ClickURL*}, where:

- AnonID: an anonymous user ID number.
- Query: the query submitted by the user in the WSE.
- QueryTime: the exact time at which the user submitted the query.
- ItemRank: if the user clicked on a result, it keeps the rank of the selected document; holds empty otherwise.
- ClickURL: The domain portion of the URL is showed as a result if the user clicked on a search result.

4.1 The Benchmark set of Related Queries

Aiming carrying out the clustering experiments, a benchmark set of related queries (*RQ*) (see *Definition 8.*) was processed randomly from AOL. To verify that the queries were partially related, each time a query was chosen, it was checked that at least existed another query, in such a way S_c was neither one nor zero. To achieve this goal, the queries with ClickURLs empty were removed due to these do not have answers associated with the queries. Furthermore, stop-words processing was previously performed before to apply S_c. The core insight is that ClickURLs allow depicting a list of retrieved documents for q (see Definition 5.). It should be noted that register with the same query q (i.e., the same terms), logged by the same user around the same time, corresponds to a single query, which was split into several registers. Providing the maximum amount of information implies to use the longest session, which at least contains one register (i.e., at least one result or document).

On the other side, it is important to highlight that AOL does not possess users' judgments. Note that the users' judgments play a fundamental role in order to know what documents are relevant for a specific query [26]. Furthermore, these relevant documents are necessary to evaluate precision, recall, and, therefore, effectiveness. To tackle this issue, users' judgments were simulated following the approach presented by [27]. Simulations of relevance judgments are presented in the following section.

4.2 Simulation of Relevance Judgments

Simulating document relevance regarding a query is not a trivial task. This task embraces a great variety of aspects, such as users' literacy [28], needed information at any one point of time, and the user's profile [29] among others. To address this problem, the approach proposed by [27], which provides the relevance probabilities for documents depending on their ItemRanks on AOL, is applied in this paper. Towards that end, $F(ItemRank(d_i), M)$ (see *Definition 6.*) is simulated using $\mathcal{M}(M, ItemRank(d_i))$ in *Definition 7.* In simple words, the relevance is assigned using a value 0 (non-relevant) or 1 (relevant), which is obtained considering the values presented in Table I (i.e, M in $F(ItemRank(d_i), M)$). The relevance probabilities were calculated using the ItemRanks, assuming the user clicks provide information about how the user interprets the query [30]. In Table I, two relevance models are presented by "AllRel" and "LastRel" columns. "AllRel" implies all clicked documents are considered as relevant; meantime "LastRel" reports that only the last clicked document is relevant. Regarding "AllRel", two variants were used for it. The first variant is named "PartialRel", which considers the individual ItemRank of each document obtained from Table 1. The second variant ("AvRel") corresponds to the average of ItemRanks of the query's recovered documents. For example, suppose three documents (d_5, d_{10}, and d_{12}) that have been recovered for a query q. The respective ItemRanks are 20, 30, and 40; therefore, the

average is 30. Subsequently, the relevance probability is determined by the "ItemRank" (average) row and the "AllRel" column, so for this example, the relevance probability for each document is 0.5106. Although the three documents have the same relevance probability, the relevance for each document is individually obtained.

Table 1: Probability of a document being relevant in the AOL dataset if it has rank k, for two different click-based relevance interpretations, as calculated by [27]

ItemRank	AllRel	LastRel
≤ 20	0.4365	0.5702
≤ 120	0.5106	0.6278
≤ 300	0.5395	0.6493
> 300	0.4705	0.3507

4.3 Clustering Experiments

Five algorithms were evaluated considering the three relevance models over the same $\widehat{D}$. Five sets of RQ were used; the smallest set of RQ contains 123 queries alongside their documents (i.e., for each document, the relevance has been assigned), meantime the biggest set comprises 2,141 queries. Aiming to compare the clusters' quality between S_C and QDSM, two well-known measures have been used, F-measure and the nearest neighbor (NN) cluster hypothesis test. F-measure was proposed by [31]; the idea behind this measure is to evaluate effectiveness in the post-processing step, in which each cluster is assigned to a class. The F-measure can be seen as a way of combining the precision and recall for a retrieval specific model, and it is defined as the harmonic mean of the model's precision and recall. In simple words, F-measure has as purpose to provide a binary classification as positive or negative according to the belonging of objects to determined classes in the clusters. F-measure allows giving more importance to precision, recall, or both. On the other hand, the nearest neighbor (NN) cluster test (which is also well-known as the (NN) test) was proposed by Voorhees ([32], [33]). In simple terms, the (NN) test reviews each of the retrieved documents for a specific query, identifying how many of its n close neighbors are relevant. The (NN) test is also used as a non-parametric classification and regression technique.

Turning towards the cluster hypothesis, QDSM should provide better effectiveness than S_c if it is possible to find more relevant documents per cluster. Each experiment was executed ten times, and results are displayed as percentages of increasing or decreasing of QDSM regarding S_c. Specifically, F-measure was used giving the same weight for precision and recall; meantime, The (NN) test was instantiated with value three in all experiments. In addition, the Students Paired t-Test (Two Samples test) was used to support the results. Five algorithms Single Link, Complete Link, Average Link, Bisection K-means, and Ward's Method, were used in each experiment.

All experiments were carried out on a server with: Intel Xeon Processor E3-1220 3.00 GHz; 16 GB Ram memory of 2133 MHz; 1 TB 7200 RPM Hard Drive; and Linux Operating System Debian Jessi 8.4.

5 Experimental Results

In this section, the quality of clusters (effectiveness) produced by QDSM and S_c is compared. To achieve this goal, the F-measure and the (NN) test were used considering the Single Link, Complete Link, Average Link, Bisection K-means, and Ward algorithms. Effectiveness was obtained using the relevance models PartialRel, AvRel, and LastRel. Note that the number of documents considered in the (NN) test corresponds to 3, it means that the relevances of the three closest documents with respect to the query were evaluated. To do that, the similarities between documents are checked alongside their relevance regarding the query. Overall, all results presented in each Table corresponds to QDSM, which are expressed in terms of percentages regarding the S_c. In Table 2, the QDSM effectiveness over the three relevance models was evaluated using the Single Link Algorithm. Note that the three relevance models were tested considering five sets of queries (# of q). From this Table, it is possible to appreciate that there is not an improvement of QDSM concerning S_c. Furthermore, Single Link presents better effectiveness for S_c than QDSM; this is consistent with the p-value obtained using The Student Paired T-test (two samples), which was 0.0029 for this Table. Generally speaking, Single Link exhibits the best results considering the "AvRel" relevance model, following by "LastRel" and finally "PartialRel" model. Continuing the same trend, the results for F-measure are exposed in Table 3. Here S_c shows again better results than QDSM, which is in line with the p-value: 0.00036, and notably the best results are presented by "AvRel" relevance model.

Regarding the Complete Link algorithm, similar results to the Single Link algorithm are presented in Tables 4 and 5, where S_c has better results than QDSM. In Table 4, the best results are provided by "LastRel" relevance model using the (NN) test, meantime that the best results using the F - measure are given by "PartialRel" model. The p-value for the (NN) test was 0.720, whilst the p-value for the F - measure was 0.00003.

On the other hand, the Average Link algorithm displays different results to Single Link and Complete Link algorithms, where QDSM is better than S_c. In Table 6, the best results are given by "PartialRel", followed by "LastRel" and "AvRel" respectively. Results showed in Table 6 are in line with the p-value: 0.00041. Following the same trend, in Table 7, QDSM presents better results than S_c for F-measure, which is in accordance with the p-value: 0.000002. It should be noted that there is no substantial difference between "AvRel" and "LastRel".

In turn, the results for the Bisection K-means algorithm are exposed in Tables 8 and 9. Both Tables provide conflicting results since Table 8 two relevance models ("PartialRel and AvRel") give good results for QDSM, meantime these relevance models provide opposing results in Table 9. These latter results are coherent with their respective p-values. The p-value for Table 8 corresponds to 0.124, whilst p-value for Table 9 is 0.904. These values are not significative due to they are not greater than 0.05.

Finally, in Tables 10 and 11, the results for the Ward's method

are displayed. Both Tables provide excellent results for QDSM in contrast to S_c (except when the number of queries is 129, using the relevance model "PartialRel"). Furthermore, the p-values associated with both Tables present the more significative value among all algorithms. The p-value associated with Table 10 is 0.00007; meanwhile, the p-value for Table 11 corresponds to 0.00001.

To sum up, considering both measures, the three relevance models, and the p-values obtained, the best results are provided by Average Link algorithm and the Ward's method.

Table 2: The (NN) test over Single Link Algorithm with *PartialRel*, *AvRel*, and *LastRel*.

# of q	*Single Link*		
	PartialRel	*AvRel*	*LastRel*
123	-30.92%	-33.73%	-42.36%
666	-11.76%	-106.36%	-69.62%
1,145	4.29%	-13.82%	4.36%
1,843	-14.70%	-22.68%	-32.44%
2,141	-40.20%	-4.74%	-11.20%

Table 3: F-measure test over Single Link Algorithm with *PartialRel*, *AvRel*, and *LastRel*.

# of q	*Single Link*		
	PartialRel	*AvRel*	*LastRel*
123	-25.15%	-12.23%	-11.89%
666	-23.96%	-44.88%	-0.35%
1,145	-22.54%	-25.10%	-5.63%
1,843	-28.60%	-43.90%	-8.04%
2,141	-20.16%	-17.20%	-6.67%

Table 4: The (NN) test over Complete Link Algorithm with *PartialRel*, *AvRel*, and *LastRel*.

# of q	*Complete Link*		
	PartialRel	*AvRel*	*LastRel*
123	-20.97%	26.76%	-41.72%
666	-14.90%	7.47%	-8.79%
1,145	-33.38%	28.44%	-29.72%
1,843	-30.80%	45.38%	-45.34%
2,141	-27.41%	41.98%	-55.13%

Table 5: F-measure test over Complete Link Algorithm with *PartialRel*, *AvRel*, and *LastRel*.

# of q	*Complete Link*		
	PartialRel	*AvRel*	*LastRel*
123	-39.50%	-11.94%	-7.34%
666	-19.09%	7.48%	-33.90%
1,145	-32.67%	-47.29%	-22.01%
1,843	-15.06%	-14.54%	-14.28%
2,141	-8.71%	-15.88%	-17.39%

Table 6: The (NN) test over Average Link Algorithm with *PartialRel*, *AvRel*, and *LastRel*.

# of q	*Average Link*		
	PartialRel	*AvRel*	*LastRel*
123	48.36%	4.50%	5.17%
666	38.47%	7.70%	18.89%
1,145	47.37%	23.69%	28.59%
1,843	47.22%	26.17%	37.29%
2,141	43.38%	32.89%	30%

Table 7: F-measure test over Average Link Algorithm with *PartialRel*, *AvRel*, and *LastRel*.

# of q	*Average Link*		
	PartialRel	*AvRel*	*LastRel*
123	8.35%	6.76%	17.37%
666	7.83%	25.27%	12.14%
1,145	2.06%	20.44%	13.39%
1,843	3.17%	11.61%	13.19%
2,141	7.38%	12.85%	22.52%

Table 8: The (NN) test over Bisection K-means Algorithm with *PartialRel*, *AvRel*, and *LastRel*.

# of q	*Bisection K − means*		
	PartialRel	*AvRel*	*LastRel*
123	15.81%	41.97%	-15.46%
666	31.73%	12.88%	-1.31%
1,145	24.22%	15.26%	-1.82%
1,843	29.48%	34.40%	-4.75%
2,141	37.98%	24.00%	-41.03%

Table 9: F-measure test over Bisection K-means Algorithm with *PartialRel*, *AvRel*, and *LastRel*.

# of q	*Bisection K − means*		
	PartialRel	*AvRel*	*LastRel*
123	-5.86%	-6.34%	5.25%
666	4.79%	-16.31%	-0.52%
1,145	-17.68%	-13.87%	2.83%
1,843	-2.50%	8.45%	7.77%
2,141	-7.82%	13.51%	12.96%

Table 10: The (NN) test over Ward's Method with *PartialRel*, *AvRel*, and *LastRel*.

# of q	*Ward′s Method*		
	PartialRel	*AvRel*	*LastRel*
123	36.47%	16.32%	39.44%
666	21.75%	8.09%	17.94%
1,145	30.61%	40.12%	28.12%
1,843	31.22%	29.21%	23.39%
2,141	21.62%	27.39%	18.80%

Table 11: F-measure test over Ward's Method with *PartialRel*, *AvRel*, and *LastRel*.

# of q	*Ward′s Method*		
	PartialRel	*AvRel*	*LastRel*
123	-4.38%	15.04%	15.37%
666	7.78%	9.34%	4.81%
1,145	8.22%	5.68%	6.85%
1,843	10.86%	7.07%	8.66%
2,141	13.08%	17.97%	8.55%

6 Discussion

The main reason traditional IR datasets were not used; it was because autonomous queries build them. As mentioned early, a traditional IR dataset is made up of a set of documents D, a set of queries Q, and a set of users' judgments JU. In a strict sense, evaluating the effectiveness considering similar queries (queries form part of the clusters) means having a set of similar queries (Q') for Q. Note that Q' must consider a set of users' judgments (JU'). Indeed, the effectiveness of two similar queries should be different. Accordingly, these datasets are not suitable for evaluating approaches based on similar queries because they have neither Q' nor JU'. On the other hand, approaches based on log files employ subject matter experts to extend and evaluate whether a document is relevant or non-relevant given a query. In short, the extension of relevant documents is common in both types of collections. In this manner, aiming to avoid using subject matter experts or extending document relevances using relevant terms, three models of relevance have been simulated in all experiments.

Concerning the effectiveness evaluation for both measures (S_c and QDSM), it is noteworthy that F-measure has been widely used in several approaches, which deal with post-retrieval clustering. Nevertheless, the use of this measure provides two drawbacks. The first one is that the result associated with this measure comprises the number of relevant and non-relevant documents related to recall and precision in its mathematical formula. Thus, the initial effectiveness changes its value once new documents are considered relevant in the post-retrieval process. The second one refers to how the clusters are conformed taking into account the different classes of objects that these contain. Consider that objects can belong to predetermine classes, and the ideal situation is given when the clusters are formed only by objects of the same class. Two terms well-known in the cluster evaluation reflect this situation, homogeneity and completeness. The idea behind homogeneity is that each cluster has few classes; meantime, completeness intends each class to be contained in a few clusters. Thus, two-cluster forming using the same objects and the same classes can have the same F-measure, while their homogeneity and completeness are different. In turn, like F-measure, the NN-test has been extensively used in several approaches to assess effectiveness. Nevertheless, this measure is not sensitive to homogeneity and completeness, since it contemplates the direct search of the n-nearest neighbors. Hence, this test is more appropriate to corroborate the cluster hypothesis, which considers the relevant documents that form the clusters.

Regarding the results presented in section "Empirical Results", it is essential to point out that there is no significant difference between values provided by F-measure and the (NN) test, excepts for the Complete algorithm (Table 4 and 5), in particular for "AvRel" where for the (NN) test, Table 4 presents favorable results for QDSM in contrast to Table 5. Besides, the p-values for both Tables differ. On the other hand, exists a substantial difference with some results presented by [1]. In particular, regarding the relevance models used in that research. There, the relevance "AllRel" is used considering the proposed by [27], meantime in this research "AllRel" has been modified by "PartialRel" and "AvRel", it means that no all documents have been considered relevant such as occurs in [1]. It is important to point out that it is unlikely that all recovered documents are relevant, such as happens in the real world. Nevertheless, the Average Link algorithm presents interesting results in both works. Concerning the results provided by the algorithms in this research, the best results are provided by Average Link and Ward algorithms using both tests (F-measure and the (NN) test). The main Ward characteristic is that it minimizes the variance of the objects belonging to a particular cluster using the "error sum of squares". In this way, each cluster should tend to have objects of a few classes (relevant and non-relevant). Carried to the hypothesis cluster context should have a clear separation between clusters of relevant documents and clusters of non-relevant documents. Therefore, the nearest closest neighbor of a relevant document should be relevant too. On the other hand, the distance (S_c or QDSM) between two clusters for the Average Link Algorithm is determined as the average distance between each object in one cluster to every object in another cluster., by which it is feasible to avoid extreme measures obtaining more homogenous clusters. The latter is in contrast with

the way to built clusters in Single and Complete algorithms. Finally, Bisection K-means is a hybrid approach between agglomerative and hierarchic clustering. This algorithm exhibits favorable results in the (NN) test except when the relevance is "LastRel", recall that in this case, only the last recovered document could be relevant.

Although the running times escape from the scope of this paper, it is worth noting that most time complexities are not high. To obtain the time complexities is necessary to consider visiting a distance matrix (i.e., one matrix for S_c and QDSM respectively) with the aim to find the n nearest-neighbor. Furthermore, calculating LCS implies to visit another matrix with M files and N rows. Note that M corresponds to a list of retrieved documents for a query q, whilst N is another list of retrieved documents for a query q'. Therefore, LCS takes $O(MN)$. Recall that LCS is used to evaluate QDSM. According to [34], the optimal implementation of Ward based on the algorithms, nearest neighbor chain, and reciprocal nearest neighbor, takes $O(N^2)$. In turn, the time complexity of Bisection K-means algorithm is $O(N^2 log_2 N)$. As for the implementation of Single Link algorithm takes $O(N^2)$ in time complexity [35]. On the other hand, Complete Link implies $O(N^2 log_2 N)$ [36]. The time complexity for the Average Link algorithm takes $O(N^2 log_2 N)$ [37]. It is important to mention that both matrixes of distances are previously built before using each clustering algorithm.

7 Conclusion and Future Work

This paper is intended to check the quality of clusters (effectiveness) built using Query-Document Similarity Measure (QDSM). To achieve this goal, the F-measure and the nearest neighbor (NN) test were used to evaluate clusters' quality. The clusters of documents were built using the AOL Query Logs Dataset. In order to provide relevance to the documents, three variants related to the ItemRanks over recovered documents were simulated. Extensive experimentation was carried out using the algorithms Single Link, Complete Link, Average Link, Bisection K-means, and Ward. According to results obtained, applying the nearest neighbor (NN) test, QDSM presents significant results using the Average Link, Ward, and Bisection K-means. On the other hand, in accordance with the results obtained by the F-measure; Ward and Average Link algorithms provide better results using QDSM than Cosine Similarity (S_C). The best results are provided by the Average Link algorithm, followed by Ward's method using QDSM, considering the three variants of relevance. Ideas for future research comprises the comparison between QDSM and other state-of-art measures.

Conflict of Interest The authors declare no conflict of interest.

Acknowledgment The Group of Smart Industries and Complex Systems (gISCOM) under grant DIUBB 195212 GI/EF supported the work presented in this article. Marco Palomino acknowledges the funding provided by the Interreg 2 Seas Mers Zeeën AGE'IN project (2S05-014).

References

[1] C. Gutiérrez-Soto, A. C. Díaz, and G. Hubert, "Comparing the effectiveness of query-document clusterings using the qdsm and cosine similarity," in 2019 38th International Conference of the Chilean Computer Science Society (SCCC), 2019, pp. 1-8.

[2] A. Mikroyannidis, "Toward a social semantic web," Computer, vol. 40, no. 11, pp. 113-115, 2007.

[3] Statista, "Average number of search terms for online search queries in the united states as of august 2017," 2017. [Online]. Available: https://www.statista.com/statistics/ 269740/number-of-search-terms-in-internet-research-in-the-us/.

[4] A. Kanavos, P. Kotoula, C. Makris, and L. Iliadis, "Employing query disambiguation using clustering techniques," Evolving Systems, vol. 11, pp. 305-315, 2020.

[5] M. Alshomary, N. Dsterhus, and H. Wachsmuth, "Extractive snippet generation for arguments, in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR 20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1969-1972.

[6] Z. A. Merrouni, B. Frinkh, and B. Ouhbi, "Toward contextual information retrieval: A review and trends, Procedia Computer Science, vol. 148, pp. 191-200, 2019, The Second International Conference on Intelligent Computing in Data Sciences.

[7] N. Jardine and C. J. van Rijsbergen, "The Use of Hierarchic Clustering in Information Retrieval," Information Storage And Retrieval, vol. 7, no. 5, pp. 217-240, 1971.

[8] C. Gutiérrez-Soto and A. Curiel Díaz, "Improving precision in IR considering dynamic environments, in Proceedings of the 21st International Conference on Information Integration and Web-Based Applications Services, ser. iiWAS2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 458-462.

[9] A. Tombros and C. J. V. Rijsbergen,"Query-sensitive similarity measures for information retrieval," Knowledge and Information Systems, vol. 6, no. 5, p. 617-642, 2004.

[10] L. Fu, D. H. lian Goh, and S. S. boon Foo, "The effect of similarity measures on the quality of query clusters," p. 396-407, 2004.

[11] K. W. Leung, W. Ng, and D. L. Lee, "Personalized concept-based clustering of search engine queries," IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 11, pp. 1505-1518, 2008.

[12] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in In Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Acm Press, 2000, pp. 407-416.

[13] S. Chawla, "A novel approach of cluster based optimal ranking of clicked urls using genetic algorithm for effective personalized web search, vol. 46, pp. 90103, 05 2016.

[14] J. Sankhavara, "Feature weighting in finding feedback documents forquery expansion in biomedical document retrieval, SN ComputerScience, vol. 1, p. 75, 2020.

[15] A. Tombros, R. Villa, and C. J. Van Rijsbergen, "The Effectiveness of Query-Specific Hierarchic Clustering in Information Retrieval, Information Processing & Management, vol. 38, no. 4, pp. 559-582, 2002.

[16] F. Raiber, O. Kurland, F. Radlinski, and M. Shokouhi, "Learning Asymmetric Co-Relevance," in Proceedings of the 2015 International Conference on the Theory of Information Retrieval, Northampton, Massachusetts, USA, 2015, pp. 281-290.

[17] S.-H. Na, "Probabilistic Co-Relevance for Query-Sensitive Similarity Measurement in Information Retrieval," Information Processing & Man- agement, vol. 49, no. 2, pp. 558-575, 2013.

[18] M. Hearst and J. O. Pedersen, "Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results, Proceedings of the 19 annual international ACM SIGIR conference on Research and development in information retrieval, 2000.

[19] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang, "Query Clustering Using User Logs," ACM Transactions on Information Systems, vol. 20, no. 1, pp. 59-81, 2002.

[20] L. Fu, D. H.-L. Goh, and S. S.-B. Foo, "The Effect of Similarity Measures on the Quality of Query Clusters," Journal of information Science, vol. 30, no. 5, pp. 396-407, 2004.

[21] R. Baeza-Yates, C. Hurtado, and M. Mendoza, "Query Recommendation Using Query Logs in Search Engines," in Proceedings of the Interna- tional Conference on Extending Database Technology. Springer, 2004, pp. 588-596.

[22] C. Gutiérrez-Soto and G. Hubert, "Evaluating the interest of revampingpast search results, in Database and Expert Systems Applications,H. Decker, L. Lhotsk a, S. Link, J. Basl, and A. M. Tjoa, Eds. Berlin,Heidelberg: Springer Berlin Heidelberg, 2013, pp. 7380.

[23] C. Gutiérrez-Soto and G. Hubert, "On The Reuse of Past Searches in Information Retrieval: Study of Two Probabilistic Algorithms, International Journal of Information System Modeling and Design (IJISMD), vol. 6, no. 2, pp. 72-92, April 2015.

[24] L. Bergroth, H. Hakonen, and T. Raita, "A survey of longest common subsequence algorithms," in Proceedings of the Seventh International Symposium on String Processing Information Retrieval (SPIRE00), ser. SPIRE 00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 39-48.

[25] G. Pass, A. Chowdhury, and C. Torgeson, "A picture of search," in Proceedings of the 1st International Conference on Scalable Information Systems, ser. InfoScale 06. New York, NY, USA: Association for Computing Machinery, 2006, p. 1-es.

[26] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B.Croft, and X. Cheng, "A deep look into neural ranking models forinformation retrieval,"Information Processing Management, p. 102067, 2019.

[27] S. Jiang, R. Zilles, and R. Holte, "Empirical analysis of the rank distribu- tion of relevant documents in web search," in In Proc. IEEE/WIC/ACM International Conf. on Web Intelligence, 2008, pp. 208-213.

[28] A. Crystal and J. Greenberg, "Relevance criteria identified by health information users during Web searches, Journal of the American Society for Information Science and Technology, vol. 57, no. 10, pp. 1368-1382, Aug. 2006.

[29] A. Spink, J. Bateman, and B. J. Jansen, "Searching the Web: a survey of EXCITE users, Internet Research, vol. 9, no. 2, pp. 117-128, May 1999.

[30] Y. Wang and E. Agichtein, "Query ambiguity revisited: clickthrough measures for distinguishing informational and ambiguous queries," in In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT 10, 2010, pp. 361-364.

[31] C. J. V. Rijsbergen, "Information Retrieval", 2nd ed. USA: Butterworth-Heinemann, 1979.

[32] Voorhees, E.M, "The effectiveness and efficiency of agglomerative hierarchic clustering in document retrieval". Ph.D. Thesis, Technical Report TR 85-705 of the Department of Computing Science, Cornell University, 1985.

[33] Voorhees, E.M, "The cluster hypothesis revisited". In Proceedings of the 8 th Annual ACM SIGIR Conference, pp. 188-196. Montreal, Canada, 1985.

[34] R. Cordeiro de Amorim, V. Makarenkov, and B. Mirkin, "A-wardp:Effective hierarchical clustering using the minkowski metric and a fastk-means initialisation,Information Sciences, vol. 370-371, pp. 343354, 2016.

[35] C. J. van Rijsbergen, "An algorithm for information structuring and retrieval, Comput. J., vol. 14, no. 4, pp. 407-412, 1971.

[36] O. Aichholzer and F. Aurenhammer, "Classifying hyperplanes in hypercubes, SIAM Journal on Discrete Mathematics, vol. 9, no. 2, pp. 225-232, 1996.

[37] H. Schtze and C. Silverstein, "Projections for efficient document clustering, SIGIR Forum, vol. 31, no. SI, pp. 74-81, Jul. 1997.