ReQue: A Configurable Workflow and Dataset Collection for Query Refinement

Mahtab Tamannaee
mtamannaee@ryerson.ca
Ryerson University, Canada

Hossein Fani
hfani@uwindsor.ca
University of Windsor, Canada

Fattane Zarrinkalam
fzarrinkalam@ryerson.ca
Ryerson University, Canada

Jamil Samouh
jtsamouh@gmail.com
Ryerson University, Canada

Samad Paydar
paydar@ryerson.ca
Ryerson University, Canada

Ebrahim Bagheri
bagheri@ryerson.ca
Ryerson University, Canada

ABSTRACT

In this paper, we implement and publicly share a configurable software workflow and a collection of gold standard datasets for training and evaluating supervised query refinement methods. Existing datasets such as AOL and MS MARCO, which have been extensively used in the literature, are based on the weak assumption that users' input queries improve gradually within a search session, i.e., the last query where the user ends her information seeking session is the best reconstructed version of her initial query. In practice, such an assumption is not necessarily accurate for a variety of reasons, e.g., *topic drift*. The objective of our work is to enable researchers to build gold standard query refinement datasets without having to rely on such weak assumptions. Our software workflow, which generates such gold standard query datasets, takes three inputs: (1) a dataset of queries along with their associated relevance judgements (e.g. TREC topics), (2) an information retrieval method (e.g., `BM25`), and (3) an evaluation metric (e.g., `MAP`), and outputs a gold standard dataset. The produced gold standard dataset includes a list of revised queries for each query in the input dataset, each of which effectively improves the performance of the specified retrieval method in terms of the desirable evaluation metric. Since our workflow can be used to generate gold standard datasets for any input query set, in this paper, we have generated and publicly shared gold standard datasets for TREC queries associated with `Robust04`, `Gov2`, `ClueWeb09`, and `ClueWeb12`. The source code of our software workflow, the generated gold datasets, and benchmark results for three state-of-the-art supervised query refinement methods over these datasets are made publicly available for reproducibility purposes.

CCS CONCEPTS

• **Information systems** → **Query suggestion**.

KEYWORDS

Gold Standard Dataset, Query Refinement, Reproducibility.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412775

ACM Reference Format:
Mahtab Tamannaee, Hossein Fani, Fattane Zarrinkalam, Jamil Samouh, Samad Paydar, and Ebrahim Bagheri. 2020. ReQue: A Configurable Workflow and Dataset Collection for Query Refinement. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412775

1 INTRODUCTION

Query refinement is a core component of many keyword-based retrieval methods, which is intended to improve retrieval performance by expanding, revising, or rewriting a given query. The objective of query refinement is to deduce the intent of the users' query and then formulate an alternative set of queries in order to fill the semantic gap between the input query and that of the documents. More recently, neural based models have received more attention for performing the supervised query refinement task [16, 25, 38]. Such approaches require high-quality training data to learn translations from the user query to an improved revised query. Different datasets such as AOL[1] and MS MARCO [2] have been adopted in the literature, which mostly include session-based search history of users. A brief summary of these datasets is included in Table 1.

One of the underlying assumptions of existing work when using a historical session-based sequence of user queries is that the last query in the session is the best refinement of the initial query, primarily because the expectation is that a user would gradually refine her query over successive attempts to find related content within the same session. However, while this assumption has not been sufficiently confirmed in the literature, neither empirically nor theoretically, it is easy to provide intuitive examples invalidating this assumption. Table 2 shows five such examples from real-world user search sessions extracted from the MS MARCO dataset where the last query clearly is not a better refinement of the initial one. For instance in the first example, the query '*what is Tanzania*' is not an appropriate refinement of the query '*is Sicily part of Italy*' since the initial query and the last query point to totally different search intents. As shown, it is probable that the user changes her intent during a session to explore information about other topics as a result of topic drift or change in search intent. For instance, in the fifth example, the user starts off by searching about *sharks* but

[1] In September 2006, a lawsuit was filed against AOL dataset in the U.S. District Court of California. Therefore, there are some ethical issues in using this dataset.
[2] https://microsoft.github.io/msmarco/

Table 1: Datasets used in the state-of-the-art for training and evaluating supervised query refinement methods.

Dataset Name	Size	Publicly Available	Citations
AOL	16M queries 3M sessions	Yes	[38], [21], [11], [1], [12], [37], [40], [30], [31], [8], [35], [7], [6], [5], [19], [34], [10], [15], [20], [11], [13]
MS MARCO	1M queries	Yes	[1], [40], [6], [5]
Yahoo Search Engine	4M queries 549K sessions	No	[25]
Tencent website	160M queries	No	[17], [16]
"Baidu Knows" Website	85K pairs of (question, best answer)	No	[27]

later switches to searching about *dogs*, which shows gradual topic drift revolving around the abstract concept of animals.

For this reason, we believe that a gold standard dataset of queries is required that would not rely on the weak assumption of gradual query improvement within the same session. Rather, each query should be paired with one or more revised queries that is guaranteed to improve retrieval performance compared to the initial query. We call such a revised query an *improved revised query*.

To produce such gold standard datasets, we propose a configurable software workflow that takes as input: (1) a dataset of queries along with their associated relevance judgements, e.g., TREC topics, (2) an information retrieval (IR) method, e.g., `BM25`, and (3) an IR evaluation metric, e.g., Mean Average Precision (`MAP`), and outputs a dataset that includes a list of improved revised queries for each of the queries in the input dataset. This is accomplished in two main steps. First, a host of state-of-the-art unsupervised query refinement techniques are implemented to systematically generate a large number of candidate queries for each input query. Second, these candidate queries are evaluated based on how they improve the performance of the given IR method, and those candidate queries that provide improvement compared to the input query are selected to be part of the gold standard query dataset.

Using this configurable software workflow, we have produced a family of gold standard datasets, collectively called ReQue (Refining Queries). We have publicly shared the code, the associated executable workflow and the ReQue datasets[3]. The advantages of our work are twofold: (1) our implementation of the proposed software workflow can be used by community members to automatically generate new gold standard datasets for any input query dataset and its associated relevance judgements, and (2) we release out of the box gold standard query datasets for `Robust04`, `Gov2`, `ClueWeb09` and `ClueWeb12` corpora and their associated TREC topics. The contributions of our work can be enumerated as follows:

(1) We propose and publicly release the source code of a configurable software workflow for automatically generating gold standard datasets for evaluating query refinement methods. The process can be easily configured based on an input query set, its associated relevance judgements, an IR method and an evaluation metric. The process produces a gold standard that ensures target revised queries improve the performance of the IR method in terms of the evaluation metric and serve the same purpose as that of the users' search intent;
(2) While our configurable workflow is able to generate any gold standard dataset, as a part of this paper, we released four gold standard datasets for each input query dataset based on `BM25`, `BM25+RM3`, `QL` and `QL+RM3` as the IR methods and `MAP` as the evaluation metric. The input datasets include the `Robust04`, `Gov2`, `ClueWeb09 (Category B)` and `ClueWeb12 (Category B)` corpora and their associated TREC topics.
(3) We apply and publicly share three strong state-of-the-art methods for supervised query refinement based on neural architectures to serve as baseline performance for the released gold standard datasets. This would improve reproducibility of the research work on the shared gold standard datasets.

[3]https://github.com/hosseinfani/ReQue/tree/cikm2020resource

2 PROPOSED WORKFLOW

In this section, we first describe our proposed configurable workflow for automatically generating gold standard datasets for the supervised query refinement task. Then, we describe how we have used this process to create the ReQue gold standard datasets. The overview of our proposed workflow for generating gold standard datasets is shown in Figure 1. The input of this workflow is a set of queries Q as input dataset and their associated relevance judgements, as well as an IR method and an evaluation metric. The output of this process is a ranked list of revised queries for each query in the input dataset, where each revised query effectively improves the performance of the IR method in terms of the given evaluation metric. The proposed workflow includes two main components: (1) *query generation*, and (2) *query evaluation*, which are laid out in detail in the following.

2.1 Query Generation

The purpose of this component is to generate a set of candidate queries that have the potential to serve as improved revised queries. The generation of candidate queries is accomplished by systematically applying a host of unsupervised query refinement techniques for each query in the input dataset. Formally, in this step, given a query $q \in Q$, a list of candidate queries C_q is generated as follows:

$$C_q = \bigcup_{s \in \mathcal{S}} s(q) \tag{1}$$

where $\mathcal{S}$ is a set of unsupervised query refinement techniques that can generate a revised query for a query q. Therefore, given a set of queries Q, the output of this step is a list of pairs $\{(q, C_q)|q \in Q\}$ each of which is a query q and its candidate revised queries C_q.

Table 2: Sample search sessions from MS MARCO dataset. For each session, the consecutive queries are enclosed in [] symbols.

#	User Session Queries
1	[is sicily part of italy]→[is sri lanka part of africa]→[what are the maldives]→ [what is great barrier reef]→[what is tanzania]
2	[immutable, definition]→[define obliged]→[meaning of industrious]→[what do vocation mean]→[legal definition capricious]→ [definition of. contempt]→[definition of famine]→[meaning of obstinate]
3	[what is google classroom]→[synonym for commotion]→[missionaries definition]→[intersect definition]→ [types of intersecting lines]→[stimulus value definition]→[definition of system unit] →[destruction of lesions definition]→ [touch definition]→[top load washer machine]
4	[definition tangible]→[define translucent]→[astringent define]→[definition of retribution]→[define defined contribution]
5	[are great white shark endangered]→[german shepherd/labrador]→[australian shepherd price]→[longevity of boston terrier]→ [cost for cairn terrier]→[is chihuahua a dog]

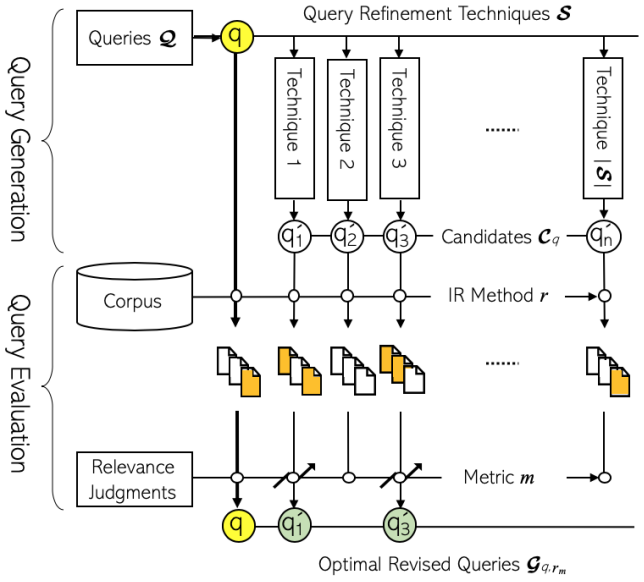

Figure 1: Overview of our proposed workflow.

We have implemented and integrated a host of query refinement techniques to represent $\mathcal{S}$ for query generation. Figure 2. shows the classification of these techniques based on the type of analysis they perform on the input query [3]. In the following, we briefly describe these techniques which are classified into two main groups: (1) *global analysis* techniques and (2) *local analysis* techniques.

2.1.1 Global Analysis Techniques. In the global analysis techniques, only the initial query terms are considered for revising the query. The techniques belonging to this group can be broadly classified into three approaches on the basis of the query terms and data sources: (1) linguistic, (2) corpus-based, and (3) web-based approaches.

Linguistic Approach. This approach uses different lexical, syntactic and semantic features of the query terms and term relationships to revise a given query. Among these features lie *word stems*, which have shown to be effective in query refinement techniques by reducing the word to its root word. We have used different algorithms including Krovetz, Lovins, PaiceHusk, Porter, Porter2, SRemoval, Trunc4, and Trunc5[4] [36] to stem the initial query terms.

[4]https://github.com/xandaschofield/stemmers

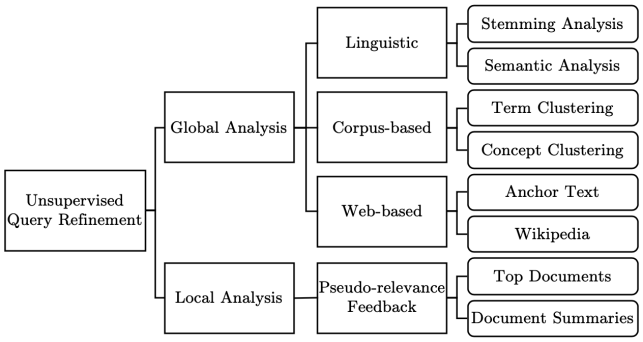

Figure 2: Classification of different unsupervised query refinement techniques used in the query generation component.

We have also included semantic analysis techniques that use an external source of linguistic knowledge, like a thesaurus, to extract related terms to the initial query terms. Given the related terms, we revise the initial query by adding the related terms to the query or replacing each query term with its identified related terms. To find the related terms, we use four different semantic analysis techniques, including: (1) WordNet-based, which works based on the synonymy relations defined in WordNet [33]; (2) ConceptNet that uses ConceptNet[5], as a freely-available relational semantic network, to understand the meaning of the query terms [18] and use the related concepts of the query terms as the related terms to refine the query; (3) A word sense disambiguation technique [39], which resolves the ambiguity of query terms by applying Pywsd[6] and then uses the synonyms of the query terms as the related terms for query refinement, and (4) Word embedding-based method that uses pre-trained word embeddings to find the most similar terms to the query terms for query refinement [23]. In our implementation, we utilize two pre-trained models including GloVe[7] and FastText[8].

Corpus-based Approach. This approach uses statistical analysis to extract the co-occurrence relationship between the words at different granularities, e.g., sentences, paragraphs and the whole

[5]http://conceptnet.io/
[6]https://github.com/alvations/pywsd
[7]https://nlp.stanford.edu/projects/glove/
[8]https://fasttext.cc/

document, in a large corpus, and then, uses these learnt relationships to revise the query terms. In our implementation, we use two possible techniques including: (1) Term Clustering [9] and (2) Concept Clustering [32].

To apply term clustering, we first build a graph based on the co-occurrence relationship between the terms mentioned in the documents of the corpus. This graph is clustered using the Louvain method [4] such that each cluster includes a set of most frequently co-occurred terms. Finally, to revise the initial query, we select the most related terms from the clusters to which the initial query terms belong. We follow the same approach to apply the concept clustering technique, however, instead of using document terms to build the graph, we first annotate the documents with the concepts defined in DBpedia, using TAGME [14]. Then, those concepts are clustered based on their co-occurrence. Further, the query is also annotated and the most related concepts from the clusters to which the initial query concepts belong are selected to revise the query. In both term clustering and concept clustering techniques, given the related terms/concepts, we revise the initial query once by adding the related terms/concepts to the query, and then by replacing each query term/concept with its related terms/concepts

Web-based Approach. The basic idea of this approach is to use web information sources to expand a given query. Two possible sources include: (1) Anchor texts [22] and 2) Wikipedia articles [2].

The anchor texts in web pages can be considered as a rich source of information for query refinement as they provide a concise summary of the content of the target page they point to. In our implementation, in order to utilize the information of anchor texts, we first learn word vectors for all the words mentioned in the Wikipedia anchor texts[9] using the word2vec model [29]. Then, given the learned vectors, the most similar terms to each query term are identified for query refinement. Furthermore, in order to utilize Wikipedia articles, we adopt the Hierarchical Category Embedding (HCE) model [26], which is a pre-trained embedding model on Wikipedia concepts that incorporates category hierarchies into concept embeddings. We refine a query by extracting the most similar concepts to each query based on the Wikipedia HCE embedding model. Finally, whether the source of data is anchor texts or Wikipedia articles, the query is revised once by adding the extracted similar terms/concepts to the initial query, and once by replacing the initial query terms with similar terms/concepts.

2.1.2 Local Analysis Techniques. The idea behind local analysis is that the terms present in the documents retrieved in response to the initial query are relevant and can be utilized to revise the initial query. Pseudo-relevance feedback is one such approach.

Pseudo-relevance Feedback Approach. This approach first retrieves the relevant documents for the initial query, and then uses these documents to compute implicit feedback, instead of explicit feedback from the user. Then, the implicit feedback is used to revise the query terms. Two possible techniques utilizing this approach include: (1) *Top documents* technique, which uses the most important terms from the top-n retrieved documents, and (2) *Document summaries* technique [24], which summarizes the retrieved documents, e.g., via clustering, and then selects the top-n terms from each cluster to be added to the initial query terms.

[9]https://wiki.dbpedia.org/downloads-2016-10#datasets

Table 3: The average number of improved revised queries for each initial query.

		Information Retrieval (IR) Method			
		BM25	BM25+RM3	QL	QL+RM3
Input Dataset	Robust04	4.25	4.81	4.06	4.39
	Gov2	2.49	2.73	2.15	2.6
	ClueWeb09	1.44	1.96	1.67	2
	ClueWeb12	1.81	2.14	1.57	2.02
	ALL	2.72	3.17	2.61	2.98

Table 4: The average MAP improvement rate (%) for each initial query.

		Information Retrieval (IR) Method			
		BM25	BM25+RM3	QL	QL+RM3
Input Dataset	Robust04	411.83	1,292.59	301.26	176.31
	Gov2	104.31	205.87	101.77	342.14
	ClueWeb09	945.22	824.70	1,751.58	1,817.08
	ClueWeb12	196.77	296.21	159.38	857.90
	ALL	467.61	783.7	652.62	778.01

Table 5: The percentage of impossible queries in each gold standard dataset.

		Information Retrieval (IR) Method			
		BM25	BM25+RM3	QL	QL+RM3
Input Dataset	Robust04	0	0	0	0.46
	Gov2	0.95	2.02	1.02	0.96
	ClueWeb09	1.80	10	3.77	4.17
	ClueWeb12	1.82	1.53	1.72	1.56
	ALL	0.83	3.13	1.26	1.59

In our implementation of both techniques, i.e., *top documents* and *document summaries*, we first use `BM25` to retrieve the related documents of the initial query from the corpus. Then, to apply the *top documents* technique, we select the terms with the highest TF-IDF scores from the top-10 retrieved documents and use these terms to expand the initial query. Similarly, to apply the *document summaries* technique, we represent each document as a TF-IDF vector and revise the initial query by first clustering the retrieved documents using Louvain method and then expanding the query by adding the most important term of each cluster to the query. While Figure 2 outlines the set of techniques that have been incorporated in the current implementation of our software workflow, it is possible to easily extend it to add new techniques in the workflow.

2.2 Query Evaluation

Given an initial query $q \in Q$, this component is intended to evaluate the candidate queries C_q generated by the query generation component, in order to select the most effective ones as the *improved revised queries*. Given the relevance judgements for each initial query, denoted by $\mathcal{R}_q$, the candidate queries are evaluated based on how they improve the performance of a given IR method

Table 6: The percentage of the best improved revised queries generated by each technique.

		Linguistic		Corpus-based		Web-based		Pseudo-relevance feedback	
		Stemming Analysis	Semantic Analysis	Term Clustering	Concept Clustering	Anchor text	Wikipedia	Top Documents	Document Summaries
Robust04	BM25	8.02	33.49	10.38	1.42	2.36	19.34	18.87	6.13
	BM25+RM3	10.14	43.32	10.6	1.84	5.07	13.82	10.6	4.61
	QL	6.07	28.5	12.15	2.8	1.87	22.9	16.82	8.88
	QL+RM3	7.44	39.53	12.09	1.86	4.19	19.07	10.7	5.12
Gov2	BM25	5.71	23.81	12.38	5.71	3.81	23.81	15.24	9.52
	BM25+RM3	9.09	41.41	5.05	4.04	3.03	28.28	3.03	6.06
	QL	4.08	22.45	8.16	6.12	2.04	27.55	18.37	11.22
	QL+RM3	10.58	37.5	11.54	5.77	5.77	19.23	5.77	3.85
ClueWeb09	BM25	1.8	32.43	10.81	0	1.8	29.73	10.81	12.61
	BM25+RM3	3.08	39.23	7.69	0.77	3.85	30	9.23	6.15
	QL	2.83	35.85	9.43	0	1.89	36.79	8.49	4.72
	QL+RM3	2.5	46.67	6.67	0.83	2.5	28.33	7.5	5
ClueWeb12	BM25	3.64	18.18	12.73	0	1.82	40	16.36	7.27
	BM25+RM3	7.69	38.46	13.85	0	3.08	26.15	4.62	6.15
	QL	3.45	34.48	13.79	1.72	1.72	29.31	12.07	3.45
	QL+RM3	6.25	42.19	6.25	1.56	1.56	31.25	7.81	3.12
ALL	BM25	5.12	28.93	11.27	1.73	2.43	26.22	15.43	8.87
	BM25+RM3	7.55	41.05	9.04	1.74	4	23.30	7.73	5.58
	QL	4.34	30.16	10.75	2.56	1.89	28.78	14.09	7.42
	QL+RM3	6.53	41.51	9.59	2.36	3.67	23.49	8.32	4.53

Table 7: The average MAP improvement (%) for each best improved revised query generated by each technique.

		Linguistic		Corpus-based		Web-based		Pseudo-relevance feedback	
		Stemming Analysis	Semantic Analysis	Term Clustering	Concept Clustering	Anchor Text	Wikipedia	Top Documents	Document Summaries
Robust04	BM25	90.66	997.01	45.29	34.14	42.84	157.41	166.11	43.67
	BM25+RM3	10,371.71	370.05	146.28	61.48	111.08	385.12	36.92	29.53
	QL	93.32	757.81	68.69	281.93	43.32	154.3	143.2	34.9
	QL+RM3	126.59	168.83	83.33	162.64	92.32	393.67	61.65	28.75
Gov2	BM25	62.28	100.61	33.93	77.36	37.28	248	46.32	21.24
	BM25+RM3	47.02	286.58	40.77	231.16	13.11	242.6	2.06	12.32
	QL	67.55	276.77	29.78	55.35	3.55	68.73	47.65	26.38
	QL+RM3	219.73	701.28	35.12	113.67	34.15	206.51	17.57	34.05
ClueWeb09	BM25	34.47	1,216.66	83.46	0	62.68	1,752.62	66.41	50
	BM25+RM3	36.26	284.12	100.06	1,965.06	782.45	2,133.92	42.56	28.95
	QL	53.63	1,269.11	172.82	0	37.29	3,579.12	64.22	62.88
	QL+RM3	38.02	991.61	291.74	5.28	85.65	4,869.17	59.4	47.33
ClueWeb12	BM25	914.81	94.89	83.82	0	13.17	269.79	148.68	60.88
	BM25+RM3	1,992.88	167.17	103.44	0	90.7	200.12	52.36	77.41
	QL	1,161.63	208.37	32.31	120	33.33	105.33	51.59	40.99
	QL+RM3	10,138.82	307.35	11.36	62.5	132.73	238.72	37.18	115.71
ALL	BM25	186.26	738.81	59.27	28.77	43.08	648.65	109.47	43.13
	BM25+RM3	4,009.31	298.63	104.34	632.94	278.995	827.81	33.27	32.52
	QL	229.07	722.326	84.91	129.69	31.65	1,107.49	87.07	41.94
	QL+RM3	1,551.56	537.79	122.26	92.88	83.72	1,610.14	48.06	47.62

with respect to an evaluation metric. Those candidates that provide the highest improvement are selected as the improved revised queries for that initial query. As a result, this component has two configuration parameters, i.e., an IR method r and an evaluation metric m. Formally, for each pair (q, C_q), given an IR method r and an evaluation metric m, a list of improved revised queries $\mathcal{G}_{q,r_m}$ is computed as follows:

$$\mathcal{G}_{q,r_m} = \{q' \in C_q | r_m(q', \mathcal{R}_q) > r_m(q, \mathcal{R}_q)\} \quad (2)$$

where $r_m(q, \mathcal{R}_q)$ is the performance of the IR method r over q, measured by the evaluation metric m, and with respect to the relevance judgments for query q, i.e., $\mathcal{R}_q$. Simply put, the elements in $\mathcal{G}_{q,r_m}$ are those queries $q' \in C_q$ for which technique r has retrieved better results in comparison to the results it has retrieved using the initial query q. The output of this step is a set of tuples $\{(q, \mathcal{G}_{q,r_m}) | q \in Q\}$ for all the queries in the input query set.

In our implementation of the query evaluation component, we have integrated Anserini [41], which provides efficient implementation of different IR methods and evaluation metrics.

3 REQUE DATASETS

To generate gold standard datasets, we applied our proposed software workflow described in Section 2 on four representative ad-hoc retrieval corpora, namely `Robust04`, `Gov2`, `ClueWeb09 (Category B)` and `ClueWeb12 (Category B)` and their associated TREC topics as input query datasets. For `Robust04`, TREC topics 301-450 and 601-650, for `Gov2`, topics 701-850, for `ClueWeb09`, topics 1-200, and for `ClueWeb12`, topics 201-300 are used as the initial queries. Further, we created a larger input query dataset, called `ALL`, by

Table 8: The average number of improved revised queries for each initial query belonging to each query type.

		Hard	Semi-hard	Semi-easy	Easy
ALL	BM25	5.71	4.26	3.27	2.55
	BM25+RM3	4.54	3.94	3.85	3.44
	QL	4.75	3.56	3.64	3.4
	QL+RM3	4.4	4.47	3.6	4.17

Table 9: The average MAP improvement rate (%) for each initial query belonging to each query type.

		Hard	Semi-hard	Semi-easy	Easy
ALL	BM25	1,588.59	109.94	46.82	19.91
	BM25+RM3	3,155.43	132.29	51.98	14.54
	QL	2,045.1	109.66	47.06	20.7
	QL+RM3	1,334.4	253.36	41.86	7.53

combining all the queries of these datasets. Further, for each input dataset, we have applied four different IR methods, namely BM25, BM25+RM3, QL, QL+RM3 and one IR metric, i.e., MAP, which resulted in a family of $4 \times 1 \times 5 = 20$ gold standard datasets. In this section, we discuss some statistics for these gold standard datasets.

The output of our workflow for each input query is a set of k improved revised queries, each of which has retrieved better results compared to the initial query in terms of MAP. Therefore, we first investigate the average value of k in the generated gold standard datasets. The greater the average value of k for a given dataset is, the richer the resulting gold standard dataset would be. For each gold standard dataset, the average value of k is reported in Table 3.

The results demonstrate that, for all the information retrieval methods, the TREC topics associated with Robust04 showed the highest improvement where the average value of k is greater than 4. This is followed by the ALL dataset where the average value of k is greater than 2.5 for all the four information retrieval methods. The richest gold standard dataset is associated with BM25+RM3 over Robust04, where for each input query, our process resulted in, on average, 4.81 improved revised queries with better MAP value compared to the original query. Further, the lowest number of average value for k was observed on ClueWeb09 and BM25 where for each input query, on average, 1.44 improved revised queries are generated. This shows that even for the weakest gold standard dataset, there is at least one improved query per input query in the generated gold standard, which can be used to train and evaluate supervised query refinement techniques.

To measure the quality of the resulting gold standard datasets for training and evaluating supervised query refinement methods, not only is the average number of improved revised queries for each initial query important, but also the amount of MAP improvement for each initial query is important. This is because it shows the effectiveness of the gold standard dataset. Given the best improved revised query for each initial query, we report the average of MAP improvement rate for each gold standard dataset in Table 4.

As shown, the minimum value of MAP improvement rate is greater than 100% for all the gold standard datasets, which means even in the worst case, the best improved revised query for an initial query almost doubled the performance of the corresponding IR method in terms of MAP. Further, there are three gold standard datasets for which the average MAP improvement rate is greater than 1,000%, meaning that, on average, the best revised query improved the MAP value of each initial query by a factor of 10. Another interesting observation is that although our proposed workflow leads to the smallest average number of improved revised queries for all the configurations of ClueWeb09 (Table 3), since the average MAP improvement rate for this input dataset is much higher than the others (Table 4), still the resulting gold standard datasets based on ClueWeb09 can be considered effective for evaluating supervised query refinement methods.

It is worth noting that in our gold standard datasets, there are some initial queries, namely *impossible queries*, whose MAP values are zero. Our software workflow has effectively generated revised queries for such impossible queries with improved MAP. However, since it is impossible to report the value of MAP improvement rate for those queries, they are excluded from our analysis in Table 4. In Table 5, the percentage of impossible queries in each gold standard dataset is reported. As it is illustrated in this table, in most gold standard datasets only less than 2% of the queries are impossible queries. Therefore, although we effectively improved their MAP value, we ignored these queries from our analysis when reporting MAP improvement rate.

As mentioned in Section 2.1, i.e., query generation, we applied a host of unsupervised query refinement methods to generate candidate queries for each initial query. To analyze the contribution of each category of unsupervised query refinement techniques on building the final gold standard dataset, we computed the percentage of best improved revised queries generated by each technique. The results are shown in Table 6. The results show that although Semantic Analysis, Wikipedia-based and Top Documents techniques have had the greatest overall contribution, almost all the unsupervised query refinement techniques have had fair contributions in at least some gold standard datasets. This supports our idea of including a host of unsupervised query refinement techniques in our proposed process, since excluding each technique results in less improvement rate for improved revised queries for at least some gold standard datasets.

To further analyze the contribution of each category of unsupervised query refinement techniques, we have reported the average MAP improvement rate for each revised query generated by each category in Table 7. Based on the results, Stemming, Wikipedia-based and Semantic Analysis have had the best performance. However, similar to our conclusion from Table 6, we can conclude that the contribution of none of the unsupervised query refinement techniques can be overlooked when generating the gold standard datasets.

To analyze how our proposed workflow performs on different types of queries, we have divided all initial queries into four equal-sized groups: hard, semi-hard, semi-easy and easy, based on their MAP value. For example, those initial queries belonging to the lowest quarter of MAP values are considered hard queries and those belonging to the highest quarter of MAP values are easy queries. For each query group, Table 8 reports the average number of improved revised queries for each initial query belonging to that query group, and Table 9 reports the average MAP improvement rate for each initial query belonging to that query type. Based on the results in

Table 10: The performance of three supervised query refinement baselines on the gold standard datasets.

		ANMT (Seq2Seq)			ACG (Seq2Seq + Attention)			HRED-qs		
		ROUGE-L	BLEU	F1	ROUGE-L	BLEU	F1	ROUGE-L	BLEU	F1
Robust04	BM25	25.68	17.84	0.268	37.46	23.01	0.385	43.25	29.51	0.451
	BM25+RM3	27.18	17.91	0.279	32.13	19.39	0.33	38.89	26.94	0.404
	QL	25.19	16.37	0.263	37.25	23.52	0.382	36.58	25.24	0.378
	QL+RM3	25.41	16.96	0.267	33.85	17.23	0.352	38.75	29.02	0.402
Gov2	BM25	19.92	12.91	0.198	47.23	29.57	0.48	25.33	18.53	0.267
	BM25+RM3	22.21	14	0.241	46.07	23.84	0.487	36.51	22.57	0.38
	QL	17.02	11.4	0.179	38.39	19.04	0.397	22.25	13.42	0.231
	QL+RM3	18.08	10.87	0.194	40.66	22.75	0.43	30.55	17.55	0.32
ClueWeb09	BM25	19.33	12.64	0.203	43.17	24.94	0.46	38.55	24.97	0.401
	BM25+RM3	10.55	7.76	0.118	44.45	27.25	0.464	19.78	13.39	0.198
	QL	15.28	1.77	0.161	48.34	25.17	0.505	33.72	24.58	0.348
	QL+RM3	16.41	9.29	0.176	41.15	21.18	0.431	29.33	17.04	0.327
ClueWeb12	BM25	18.55	13.42	0.186	50.87	30.5	0.531	46.87	30.35	0.5
	BM25+RM3	24.47	16.02	0.239	50.41	33.32	0.534	36.08	20.53	0.404
	QL	12.56	6.06	0.156	43.48	2.59	0.479	24.37	12.78	0.298
	QL+RM3	24.86	15.91	0.253	49.45	24.9	0.513	24.54	8.54	0.257
ALL	BM25	28.61	19.47	0.299	38.01	24.4	0.39	42.84	31.01	0.442
	BM25+RM3	28.65	18.36	0.297	35.17	22.16	0.363	41.17	29.98	0.426
	QL	24.97	17.6	0.262	34.99	23.36	0.368	39.3	29.74	0.405
	QL+RM3	25.76	10.84	0.265	35.99	21.93	0.372	41.63	28.96	0.425

both tables, it can be concluded that the harder the initial query is, the more successful our workflow will be in providing revised queries with higher MAP values. Indeed, an initial easy query includes enough information to retrieve the related documents from the corpus and it is more challenging to refine it to get better results.

4 ESTABLISHING BENCHMARKS ON REQUE

In order to establish query refinement baselines for the set of released ReQue gold standard datasets, we have adopted three state-of-the-art supervised methods for query refinement that are based on Recurrent Neural Network (RNN) architectures. Given an input query, the trained query refinement model suggests a list of improved revised queries. The three methods are as follows:

ANMT (Seq2Seq) [28] is an RNN-based encoder-decoder architecture trained to take a sequence of queries, and generate a corresponding sequence of suggested queries. In this model, the encoder is a bidirectional RNN that learns the representation of the input query sequence in two directions: left-to-right (forward pass) and right-to-left (backward pass). The concatenation of the resulting forward and backward hidden states creates the encoder hidden states representing the query-level encoded information of the input query sequence. The decoder is a unidirectional RNN that decodes the context vector to generate the output query.

ACG (Seq2Seq + Attention) [12] is proposed to improve Seq2Seq models by applying a global word-level attention mechanism on top of the Seq2seq model, to learn the weights between the terms of the initial query and the target suggested query. The word-level attention mechanism dynamically changes the context vector by assigning weights to the hidden states of the encoder during the decoding process.

HRED-qs [38] is based on an end-to-end hierarchical recurrent encoder-decoder architecture. This architecture encodes the query session information at two levels: query level and session level. Similar to Seq2Seq models, the query encoder is a bidirectional RNN, which learns the query term sequence in two directions. The session encoder is a unidirectional RNN, which encodes a session as a sequence of queries. The search intent is formulated using query encoding and its query session encoding. Finally, in the decoding process, an RNN decoder is used for generating the next query based on the results of the query-level and session-level encoders.

4.1 Results

We have conducted an experiment to show how the three above-mentioned baselines perform on the ReQue datasets. For each gold standard dataset belonging to ReQue, given the pairs $\{(q, q')|q' \in \mathcal{G}_{q,r_m}\}$,we randomly selected 70% for training, 15% for validation and 15% for testing, with no overlapping as suggested by Ahmad et al. [1]. After running each model for 100 epochs, we have reported their performance on the test set in terms of ROUGE-L, BLEU and F1 in Table 10. Based on table, although these baselines perform differently on different gold standard datasets, in all the cases, ACG and HRED-qs outperform ANMT in terms of all the evaluation metric, and in most cases, ACG that uses the attention mechanism to improve the simple Seq2Seq model outperform HRED-qs. The similar observation is also reported in [1] and [12].

5 CONCLUDING REMARKS

In this paper, we proposed a configurable workflow to generate gold standard datasets for training and evaluating supervised query refinement methods. Given an input query dataset, this process

includes two main components, i.e., *query generation* and *query evaluation*. In the query generation component, a host of unsupervised query refinement approaches are utilized to generate candidate queries for each query in the input dataset. Then, given an information retrieval method and an evaluation metric, the query evaluation component selects those candidate queries that effectively improve the performance of the retrieval method in terms of the given evaluation metric. To provide gold standard datasets, we configured our workflow based on `BM25`, `BM25+RM3`, `QL` and `QL+RM3` retrieval methods and the `MAP` evaluation metric and applied the proposed workflow on four representative ad-hoc retrieval corpora, namely `Robust04`, `Gov2`, `ClueWeb09` and `ClueWeb12` and their associated TREC topics as input. The resulting gold standard datasets, named ReQue, and the source code of our developed process are publicly released. Additionally, we reported the benchmark results for a set of three state-of-the-art supervised query refinement methods on ReQue datasets for future reproducibility purposes.

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context Attentive Document Ranking and Query Suggestion. In *42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*. 385–394.

[2] Bashar Al-Shboul and Sung-Hyon Myaeng. 2014. Wikipedia-based query phrase expansion in patent class search. *Information Retrieval* 17, 5-6 (2014), 430–451.

[3] Hiteshwar Kumar Azad and Akshay Deepak. 2019. Query expansion techniques for information retrieval: A survey. *Inf. Process. Manag.* 56, 5 (2019), 1698–1735.

[4] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks.

[5] Fei Cai and Honghui Chen. 2017. Term-level semantic similarity helps time-aware term popularity based query completion. *J. Intell. Fuzzy Syst.* 32, 6 (2017), 3999–4008.

[6] Fei Cai and Maarten de Rijke. 2016. Learning from homologous queries and semantically related terms for query auto completion. *Inf. Process. Manag.* 52, 4 (2016), 628–643.

[7] Fei Cai and Maarten de Rijke. 2016. Selectively Personalizing Query Auto-Completion. In *39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR*. 993–996.

[8] Fei Cai, Shangsong Liang, and Maarten de Rijke. 2016. Prefix-Adaptive and Time-Sensitive Personalized Query Auto Completion. *IEEE Trans. Knowl. Data Eng.* 28, 9 (2016), 2452–2466.

[9] Claudio Carpineto, Renato de Mori, Giovanni Romano, and Brigitte Bigi. 2001. An information-theoretic approach to automatic query expansion. *ACM Trans. Inf. Syst.* 19, 1 (2001), 1–27.

[10] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2017. Personalized Query Suggestion Diversification. In *40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 817–820.

[11] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2018. Attention-based Hierarchical Neural Query Suggestion. In *41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR*. 1093–1096.

[12] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. In *2017 ACM on Conference on Information and Knowledge Management*. 1747–1756.

[13] Heng Ding, Shuo Zhang, Darío Garigliotti, and Krisztian Balog. 2018. Generating High-Quality Query Suggestion Candidates for Task-Based Search. In *40th European Conference on IR Research, ECIR (Lecture Notes in Computer Science, Vol. 10772)*. Springer, 625–631.

[14] Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *19th ACM Conference on Information and Knowledge Management, CIKM 2010*. ACM, 1625–1628.

[15] Nicolas Fiorini and Zhiyong Lu. 2018. Personalized neural language models for real-world query auto completion. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*. Association for Computational Linguistics, 208–215.

[16] Fred X. Han, Di Niu, Haolan Chen, Kunfeng Lai, Yancheng He, and Yu Xu. 2019. A Deep Generative Approach to Search Extrapolation and Recommendation. In *25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*. 1771–1779.

[17] Fred X. Han, Di Niu, Kunfeng Lai, Weidong Guo, Yancheng He, and Yu Xu. 2019. Inferring Search Queries from Web Documents via a Graph-Augmented Sequence to Attention Network. In *The World Wide Web Conference, WWW*. 2792–2798.

[18] Ming-Hung Hsu, Ming-Feng Tsai, and Hsin-Hsi Chen. 2006. Query Expansion with ConceptNet and WordNet: An Intrinsic Comparison. In *Information Retrieval Technology, Third Asia Information Retrieval Symposium, AIRS 2006 (Lecture Notes in Computer Science, Vol. 4182)*. Springer, 1–13.

[19] Zhipeng Huang and Nikos Mamoulis. 2017. Location-Aware Query Recommendation for Search Engines at Scale. In *15th International Symposium, SSTD (Lecture Notes in Computer Science, Vol. 10411)*. 203–220.

[20] Aaron Jaech and Mari Ostendorf. 2018. Personalized Language Model for Query Auto-Completion. In *56th Annual Meeting of the Association for Computational Linguistics, ACL*. Association for Computational Linguistics, 700–705.

[21] Jyun-Yu Jiang and Wei Wang. 2018. RIN: Reformulation Inference Network for Context-Aware Query Suggestion. In *27th ACM International Conference on Information and Knowledge Management, CIKM*. 197–206.

[22] Reiner Kraft and Jason Y. Zien. 2004. Mining anchor text for query refinement. In *13th international conference on World Wide Web, WWW 2004*. ACM, 666–674.

[23] Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query Expansion Using Word Embeddings. In *25th ACM International Conference on Information and Knowledge Management, CIKM 2016*. ACM, 1929–1932.

[24] Kyung-Soon Lee, W. Bruce Croft, and James Allan. 2008. A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008*. ACM, 235–242.

[25] Ruirui Li, Liangda Li, Xian Wu, Yunhong Zhou, and Wei Wang. 2019. Click Feedback-Aware Query Recommendation Using Adversarial Examples. In *The World Wide Web Conference, WWW 2019*. ACM, 2978–2984.

[26] Yuezhang Li, Ronghuo Zheng, Tian Tian, Zhiting Hu, Rahul Iyer, and Katia P. Sycara. 2016. Joint Embedding of Hierarchical Categories and Entities for Concept Categorization and Dataless Classification. In *COLING 2016, 26th International Conference on Computational Linguistics, The Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*. ACL, 2678–2688.

[27] Xiaoyu Liu, Shunda Pan, Qi Zhang, Yu-Gang Jiang, and Xuanjing Huang. 2018. Generating Keyword Queries for Natural Language Queries to Alleviate Lexical Chasm Problem. In *27th ACM International Conference on Information and Knowledge Management*. 1163–1172.

[28] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *2015 Conference on Empirical Methods in Natural Language Processing, EMNLP*. The Association for Computational Linguistics, 1412–1421.

[29] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. 3111–3119.

[30] Bhaskar Mitra. 2015. Exploring Session Context using Distributed Representations of Queries and Reformulations. In *38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3–12.

[31] Bhaskar Mitra and Nick Craswell. 2015. Query Auto-Completion for Rare Prefixes. In *24th ACM International Conference on Information and Knowledge Management, CIKM*. 1755–1758.

[32] Apostol Natsev, Alexander Haubold, Jelena Tesic, Lexing Xie, and Rong Yan. 2007. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *15th International Conference on Multimedia*. ACM, 991–1000.

[33] Dipasree Pal, Mandar Mitra, and Kalyankumar Datta. 2014. Improving query expansion using WordNet. *J. Assoc. Inf. Sci. Technol.* 65, 12 (2014), 2469–2478.

[34] Dae Hoon Park and Rikio Chiba. 2017. A Neural Language Model for Query Auto-Completion. In *40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1189–1192.

[35] Shuyao Qi, Dingming Wu, and Nikos Mamoulis. 2016. Location Aware Keyword Query Suggestion Based on Document Proximity. *IEEE Trans. Knowl. Data Eng.* 28, 1 (2016), 82–97.

[36] Alexandra Schofield and David M. Mimno. 2016. Comparing Apples to Apple: The Effects of Stemmers on Topic Models. *Trans. Assoc. Comput. Linguistics* 4 (2016), 287–300.

[37] Milad Shokouhi. 2013. Learning to personalize query auto-completion. In *36th International ACM SIGIR conference on research and development in Information Retrieval*. 103–112.

[38] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. In *24th ACM International Conference on Information and Knowledge Management, CIKM 2015*. ACM, 553–562.

[39] Liling Tan. [n.d.]. Pywsd: Python Implementations of Word Sense Disambiguation (WSD) Technologies [software]. https://github.com/alvations/pywsd.

[40] Stewart Whiting and Joemon M. Jose. 2014. Recent and robust query auto-completion. In *23rd International World Wide Web Conference, WWW*. 971–982.

[41] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1253–1256.