Leveraging Semantic and Lexical Matching to Improve the Recall of Document Retrieval Systems: A Hybrid Approach

Saar Kuzi[*]
University of Illinois at Urbana-Champaign
skuzi2@illinois.edu

Mingyang Zhang
Google Research
mingyang@google.com

Cheng Li
Google Research
chgli@google.com

Michael Bendersky
Google Research
bemike@google.com

Marc Najork
Google Research
najork@google.com

ABSTRACT

Search engines often follow a two-phase paradigm where in the first stage (the *retrieval* stage) an initial set of documents is retrieved and in the second stage (the *re-ranking* stage) the documents are re-ranked to obtain the final result list. While deep neural networks were shown to improve the performance of the re-ranking stage in previous works, there is little literature about using deep neural networks to improve the retrieval stage. In this paper, we study the merits of combining deep neural network models and lexical models for the *retrieval* stage. A hybrid approach, which leverages both semantic (deep neural network-based) and lexical (keyword matching-based) retrieval models, is proposed. We perform an empirical study, using a publicly available TREC collection, which demonstrates the effectiveness of our approach and sheds light on the different characteristics of the semantic approach, the lexical approach, and their combination.

1 INTRODUCTION

The ad hoc retrieval task is commonly addressed using a two-phase approach. In the first stage (the retrieval stage), an initial result list of documents is retrieved from the collection for the query. Then, in the second stage (the re-ranking stage), the initial result list is re-ranked to generate the final list. The focus of this work is on the retrieval stage where the main goal is to maximize the recall of the relevant documents retrieved. This is different than the goal of re-ranking which is to optimize the precision at high ranks of the final list. Furthermore, since the retrieval stage is performed against all documents in the collection, a major requirement from a model is to be efficient. The common practice for the retrieval stage is to use a lexical-based model, such as BM25 [36]. A lexical model assigns a relevance score to a document with respect to a query relying on the level of matching between the query and the document terms. This type of model is likely to achieve a reasonable level of recall since the occurrence of the query words in documents is often a necessary condition for relevance. The lexical retrieval approach is also efficient due to the use of an inverted index.

A retrieval that relies only on a lexical model is likely to be non-optimal. For example, such a model would have difficulty in retrieving relevant documents that have none of the query terms. This problem is partially a vocabulary mismatch problem in which a relevant document uses terms that are related to but different from the query terms. Furthermore, relying solely on keyword matching may also not align well with people's actual information needs. When people search, what often they truly care about is whether the search results can address their needs, rather than whether the results contain the query words.

To illustrate this point, an example query from our evaluation data set is presented in Table 1. In the table, we can see a passage from a relevant document retrieved using BM25 and a passage from a relevant document retrieved by the semantic model we used in this paper. We can see that while the lexical document contains the query term "fatality", the semantic document contains a related term "kill". A further examination of the document revealed that the term "fatality" does not appear in any part. Thus, using a semantic model we can retrieve relevant documents that cover only some of the query terms.

The main idea of semantic matching of text is that it does not rely heavily on exact keyword matching. Instead, it measures complex relationships between words to capture semantics. Effective semantic models in recent years were mostly learned using deep neural networks [14]. Deep neural networks also attracted great interest in the IR community and many approaches for the re-ranking stage were devised [35]. The common main idea of the works on the subject is to use a large amount of training data, leveraging either query logs or weak supervision, to learn a model for the prediction of relevance between a document and a query. These works often follow the standard two-phase retrieval paradigm in which the retrieval stage is executed using a lexical-based model, and the result list is re-ranked using a neural network model.

The study of semantic models for the retrieval stage is a subject that was rarely studied in previous works. Two possible reasons for this can be: (1) semantic models tend to have lower recall due to their soft matching nature, and (2) before the recent development of fast approximate KNN search [18], using neural networks for retrieval had a very high cost. This is because running a query through a neural model and pairing it with each of the documents in the collection is extremely inefficient.

In this work, we study the effectiveness of semantic models for the retrieval stage. Our main premise is that even if the recall of the semantic retrieval is low, it still can retrieve relevant documents not covered by the lexical model. This is a reasonable assumption due to the complementary nature of the two approaches. Thus, to benefit from both approaches, we propose a lexical-semantic hybrid retrieval approach. The main idea is to run a semantic and lexical

[*] This work was done while interning at Google.

retrieval in parallel and merge the two result lists to create the initial list for re-ranking. Since the retrievals can be performed in parallel, our approach can be efficiently used in any system.

Besides the difference at which stage (retrieval vs. re-ranking) the model is used, another major difference between our model and many of the previously proposed neural models for IR [21, 37, 40] is that our model does not require access to large-scale query logs. Inspired by the recent development of pre-trained language models [14], we design weakly supervised learning tasks to learn corpus-specific semantics. This makes our model useful to learn domain-specific knowledge for a new search scenario and for systems where logs cannot be collected.

The suggested approach is deployment achievable for the following reasons: (1) the approach relies on adding a second retrieval source and is thus not expected to hurt the performance of the current lexical-based approach, (2) the neural model training is weakly supervised and no training data is in need, (3) an approximate KNN search is used which is very efficient and is not expected to affect the system latency, and (4) our method is fully implemented using open-source software and can be thus easily reproduced.

An extensive empirical analysis of the proposed approach is performed using a public TREC collection. The analysis confirms that the semantic approach can retrieve a large number of relevant documents not covered by the lexical approach. Then, we show that by using a simple unsupervised approach for merging the result lists, significant improvements in the recall can be achieved. Finally, an exploration of the different characteristics of the semantic and lexical retrieved documents is performed, using both quantitative and qualitative measures, that sheds light on the complementary nature of the two approaches.

To summarize, the main contributions of this work are:

- Proposing and studying a novel hybrid document retrieval approach that leverages lexical and semantic (neural network-based) models. The proposed approach is efficient enough to be deployed in any commercial system.
- Proposing an effective end-to-end weak supervision training approach for the retrieval stage that does not rely on any external resources.
- Conducting an empirical evaluation that demonstrates the effectiveness and robustness of the suggested approach compared to the lexical-only approach.
- Conducting an empirical study that illustrates the different characteristics of the lexical model, the semantic model, and their combination.

2 RELATED WORK

The main novelty of our work is that we study a lexical-semantic hybrid approach to improve the recall of the retrieval stage. While there has been a large body of work in the area of neural information retrieval (e.g., [15, 21, 35, 37, 40]), the focus was mainly on improving the re-ranking precision.

Semantic retrieval approaches that do not rely on deep neural networks were proposed in some previous works. In one line of works [3, 8], Latent Semantic Indexing (LSI) was used to generate dense representations for queries and documents which were either used alone for retrieval or combined with a lexical approach. The suggested approaches, however, demonstrated the limited ability of LSI in improving the effectiveness of the retrieval stage. In another work [6], KNN search was used for semantic retrieval by leveraging a statistical translation model. In this work, our focus is on studying neural network-based approaches.

Table 1: An example of relevant documents retrieved by the lexical and the semantic approaches. Only a part of the document which contains the relevant information is presented.

Query: “Weather Related Fatalities”
Information Need: A relevant document will report a type of weather event which has directly caused at least one fatality in some location.
Lexical Document
“... Oklahoma and South Carolina each recorded three fatalities. There were two each in Arizona, Kentucky, Missouri, Utah and Virginia. Recording a single lightning death for the year were Washington, D.C.; Kansas, Montana, North Dakota, ...”
Semantic Document
“... Closed roads and icy highways took their toll as at least one motorist was killed in a 17-vehicle pileup in Idaho, a tour bus crashed on an icy stretch of Sierra Nevada interstate and 100-car string of accidents occurred near Seattle ...”

There have been some previous works on developing neural network-based semantic approaches for the retrieval stage of documents. One work [46] proposed a model that learns sparse vectors for documents and queries which can be used for retrieval with an inverted index. In another work [19], KNN search was used for the retrieval stage with neural network-based embeddings. The suggested approach [19], however, is not applicable for large collections since it requires the learning of document-specific representations for the entire collection. In our paper, the focus is on studying the integration of lexical and neural approaches in the general case. Thus, our approach can be applied on top of any semantic model to further improve its performance. Furthermore, the approach we take in this paper uses an existing neural model with some small modifications, whereas in those previous works new models were designed for the task. For this reason, our approach can more easily leverage novel neural models in the future.

A lexical-semantic hybrid approach was previously studied for the re-ranking stage [26]. Specifically, two neural networks were trained jointly accounting for local (term-based interactions) and distributed (semantic) representations of queries and documents. In this work, we show that a hybrid approach can also help to increase the recall of the retrieval stage.

The recent success of applying the pre-trained language model BERT [14] to many NLP tasks motivated the development of several BERT-based re-ranking models for IR [31, 33, 45]. The main idea of these works is to treat the query and the document as two consecutive sentences in BERT and use feed-forward layers on top of BERT’s classification layer to compute the relevance score. This approach was used for re-ranking of passages [31, 33], and more recently to re-rank news-wire documents [45]. Motivated by the success of BERT for the re-ranking task, in this work, we use the BERT architecture for retrieval. Differently from previous works,

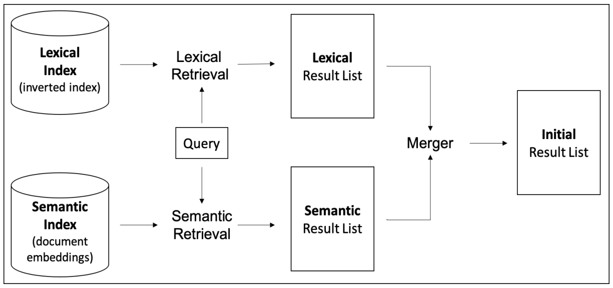


Figure 1: The hybrid retrieval approach.

we take a representation-based approach, by generating embedding vectors, which is more applicable for the retrieval stage.

Neural network-based semantic retrieval models were already applied to several other applications rather than document retrieval. In one work [11], BERT was used for weighting terms in the inverted index of passages. In another work [27], an efficient neural re-ranking and retrieval approach was suggested by assuming independence between query terms. This approach [27], however, was mainly studied for the re-ranking of passages. Finally, neural models were shown to be more effective than lexical models for the retrieval stage in QA systems, conversational agents, and product search [2, 23, 30, 42].

Recall can also be improved through query expansion [9]. This approach, however, is often not used in commercial systems due to efficiency issues. First, query expansion uses very long queries which result in a prohibitive query evaluation time [4, 22, 39]. Second, the most effective approach, which relies on the result list to learn expansion terms (pseudo-relevance feedback) [1, 41], requires two sequential retrieval steps and is thus not efficient enough.

Document expansion is another technique that is used for improving the recall of retrieval systems [38]. Recent works have demonstrated the effectiveness of this approach for the retrieval of passages [32, 34]. Using it for document retrieval, however, was shown to have limited effectiveness [5].

3 A HYBRID RETRIEVAL APPROACH

In this paper, the focus is on the retrieval stage where the goal is to retrieve an initial set of documents of size c using both semantic and lexical models. The next step, which is out of the scope of this research, is the re-ranking stage in which the initial result list is ranked to generate a final list of size c' (usually, $c' \ll c$).

The hybrid approach is depicted in Figure 1. The approach requires the existence of two indexes: (1) a lexical index (an inverted index), and (2) a semantic index (document embeddings matrix). Given a query q, two retrieval steps are performed in parallel. Lexical retrieval is performed in which the words in the query are matched with the words in documents. In this paper, we use the BM25 model [36] which is a classical retrieval approach that is highly effective and widely used by current retrieval systems. (For example, BM25 is the main approach taken by systems in recent IR competitions [10].) Semantic retrieval is also performed by first inferring an embedding vector for the query and then performing KNN search against the semantic index. The two result lists, each of size c, are pooled and then a merger is used to select c documents from the pool to obtain the initial result list.

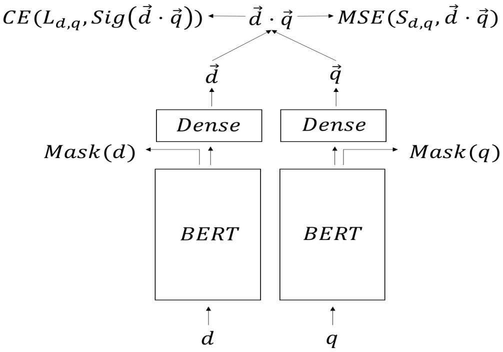


Figure 2: The neural network architecture of the semantic retrieval model.

The hybrid retrieval approach was developed to be efficient enough so that it could be deployed in any system. Our main goal is to avoid any extra overhead on top of the lexical (inverted index-based) approach which is the standard in current systems. The hybrid approach, by using two independent retrieval stages (semantic and lexical), can achieve this goal since the two can be performed in parallel. Furthermore, since we use approximate KNN search for the semantic retrieval [16, 17, 28], it is expected to be as efficient as an inverted index-based search [6, 24].

In the remainder of this section, we cover the technical details regarding the implementation of the hybrid approach including details about the semantic retrieval implementation as well as the merging step.

3.1 Semantic Retrieval

This section describes the details of the neural model used for the semantic retrieval part. It is important to mention that in this work we are not interested in the full optimization of the semantic (neural) model but to study the potential benefits of combining semantic and lexical result lists. To that end, we make implementation decisions mainly in light of the findings of recent works on language understanding to obtain a sufficiently effective semantic model. Studying the effectiveness of different semantic models for the hybrid approach is left for future work.

3.1.1 Neural Model Architecture. The main idea of semantic retrieval is to generate query and document embedding vectors. Then, at serving time, a semantic similarity between a query and a document can be measured using the cosine function. The general architecture of the neural network, which was used for the semantic model, is depicted in Figure 2. To generate query/document embeddings, we adopt the early idea of Siamese neural network architectures [7]; this architecture was selected since it enables us to obtain query-independent document representations for indexing. Specifically, we are given a neural model that gets as an input a sequence of words and outputs a continuous vector. This model is used to generate both query and document vectors in parallel. In this paper, the architecture of the BERT model was used [14].

We chose this model as it was shown to achieve state-of-the-art performance in many NLP tasks. To generate an embedding vector for a document/query, we collect the pooled output from BERT and add an extra dense layer on top of it. The parameters of the BERT module are shared by the query and the document model to learn the common knowledge in the text. The parameters of the top dense layers of the query and the document model are trained separately so that we can learn query- and document-specific representation. Then, the dot product between the vectors serves as a predicted *relevance score* of the document to the query. The loss function for a pair of a query q and a document d, which is associated with a binary relevance label $L_{d,q}$ and a continuous relevance score $S_{d,q}$, is defined as:

$$\mathcal{L} = CE(L_{d,q}, Sigmoid(\vec{d}\cdot\vec{q})) + MSE(S_{d,q}, \vec{d}\cdot\vec{q}) + Mask(q) + Mask(d).$$

Where CE and MSE are the Cross Entropy loss and Mean Squared Error loss, respectively; $Mask(\cdot)$ is the masked language model loss used in BERT; $\vec{q}$ and $\vec{d}$ are the vectors generated by the neural model. We use the two losses as it is expected for the two to be complementary. While the *CE* loss can help learn the rough distinction between something that is completely non-relevant to something that is somehow relevant. The *MSE* loss can fine-tune the model to be more discriminative. We tried to fine-tune the model with just *CE* and *MSE* loss at the end of the training process but didn't notice much difference. Probably this is because differently from the original BERT paper, here we are directly training a model on the target data set.

3.1.2 Training data. Semantic retrieval models, learned using deep neural networks, require large amounts of training data which is often hard to obtain. To address this issue, several previous works have explored using weak supervision for the re-ranking task [13, 20, 29]. In this work, we also use weak supervision and demonstrate its effectiveness for the retrieval stage. Furthermore, unlike previous works, we propose an end-to-end training data generation pipeline that does not rely on any auxiliary resources. Generalizing the results obtained in this work to semantic models that were learned using labeled data is an important direction worth exploring in future research (when such data is available). Our proposed framework is general enough to facilitate the study of this direction.

To obtain training queries, tri-grams and bi-grams that appear in at least 5 documents in the collection are extracted. Then, queries with less than 10 results when using BM25 are filtered out to make sure that we have enough training data for learning effective representations. Next, document-query pairs, associated with a relevance score and a binary relevance label, are generated using a weak supervision approach (similarly to a previous work [13]). For each query, 10 documents are retrieved using BM25 and each document is replaced by at most 5 passages from it.[1] Only passages that contain all query terms are used. We use passages instead of using the entire document due to the limitation of BERT in handling long sequences of words [12]. The query-document pairs, which are generated using our approach, are considered relevant. Non-relevant pairs are generated using random sampling. To create relevance scores for query-document pairs, we randomly remove query terms from a relevant passage and replace them with random terms from the vocabulary. Specifically, a pair of a bi-gram query and a relevant passage will be transformed into three pairs by adding two more pairs where the passage only matches a single term. To determine the match score, any relevance measure score like BM25 can be used. In practice, we found that using predefined scores works pretty well. That is, the full match score is set to 1, while the partial matching score is set to 0.6. Similarly, a pair of a tri-gram query and a relevant passage will be transformed into seven pairs; the full match score will be 1, while the partial matching score will be 0.55 and 0.65 for single and double matching, respectively.

3.1.3 Retrieval. After the model was learned, it can be used to generate the semantic index by inferring vectors for all passages in the collection in an offline manner. Then, at serving time, KNN search can be used for semantic retrieval. Since we have passage embeddings rather than document embeddings, there is a need to transform the result list to the document level. To do that, we sum up the scores of retrieved passages per document to obtain a document score.

3.2 Hybrid Merging

In this step, the documents retrieved by the semantic and the lexical approach are pooled to create a document set of size up to $2c$. Then, a merger function assigns a score to every document in the pooled set. Finally, c documents with the highest scores are used to form the initial result list.

Using either the lexical or the semantic model as the merger function is likely to favor documents from only one of the two models. This is not desirable since we are interested in having both semantic-based and lexical-based relevant documents in the final list. When using neural networks for re-ranking, previous works tended to rely on semantic scores because their retrieval stage has already enforced lexical matching (e.g., [13, 15]). For the retrieval stage, however, relying on semantic scores may not be the best choice. One reason for this is that to generate semantic scores for the documents returned by the lexical approach, we need to run them against the neural model which may be inefficient. Furthermore, our preliminary examinations showed that the relevant documents in the semantic result list do not necessarily appear in high ranks. This suggests that semantic retrieval is not as discriminative as a lexical one. This is probably because embeddings can be regarded as smoothed representations of text and are hence not discriminative enough. On the one hand, they are strong at finding semantically similar text; on the other hand, facing a piece of semantically matched text and a piece of exactly matched text, as their smoothed representations would be quite similar, just relying on semantic representations to rank them may not be very effective.

To address those issues, we use the relevance model RM3 [1] as a merger which was shown to be an effective approach for TREC-style documents in some previous works (e.g., [44]). RM3 is essentially a probability distribution induced from the top documents in the initial result list and the original query which is supposed to serve as a representation of the user's information need; we refer the reader to the original paper [1] for more details about this model. RM3 is used as a merger in the following way. First, an RM3 model is induced from the result list of the lexical model (we use the lexical results since semantic scores are not as discriminative as lexical

[1] A document is split into passages of 20 words with a sliding window of size 10.

scores). Then, each document in the pooled set is scored using the RM3 model. Finally, the c documents with the highest scores are selected to form the initial result list. Using RM3 is advantageous in this scenario because it takes into account the lexical similarity between the query and the document as well as the similarity between the document and related terms which can be indicative of semantic similarity. We note that other approaches for the merger step can also be used. Yet, as will be shown in the experimental section, using RM3 already results in significant improvements and is simple and easy to implement. From a practical point of view, it is important to mention that since we use RM3 only to score the documents in the pooled set, the query processing time is not supposed to increase largely. This is contrary to the common use of RM3 for pseudo-relevance feedback which requires two independent retrieval steps.

4 EVALUATION

4.1 Experimental Setup

4.1.1 Data set. A TREC collection (disks 1&2) of 441,676 news-wire documents was used for the evaluation. The titles of TREC topics 51-200 served as queries. This collection was selected since our focus is on performing a systematic analysis of retrieval models that rely solely on textual data. Thus, we are interested in a collection that has minimal noise and that contains reliable relevance judgments. Since the focus in this work is on weak supervision-based semantic models, our method does not require large data sets of labeled data, and we thus leave the evaluation on such data sets (for example, the TREC DL data set [10]) for future work.

Using this collection, our training data set ended up having 3.8M bi-gram queries, 1.7M tri-gram queries, and about 1B training examples (passage-query pairs). As already mentioned in the previous section, we split the documents in the collection into passages, resulting in approximately 22M passages. Thus, to generate an effective result list of documents, a large enough number of passages is needed to obtain enough evidence regarding each document. In this paper, we empirically set this value to 10,000.

4.1.2 Lexical model implementation. BM25 was used as the lexical model (denoted **Lexical**). The Anserini toolkit [43] was used for document and query pre-processing and for the implementation of the BM25 model (used as a baseline or as part of the hybrid approach) and of the RM3 model (used in the merging step of the hybrid approach). RM3 was not used as a baseline since it requires two consecutive retrieval steps and is thus not applicable to many search applications. The free parameters of the lexical approaches were set to default values.[2] One of the reasons for choosing Anserini is that its default free parameters for the lexical models are tuned to produce highly effective results for TREC collections [43]. Krovetz stemming and stopword removal were applied to both queries and documents. For the evaluation, only queries for which all query terms are in the vocabulary of the semantic model were used (121 queries). We limited the evaluation to these queries to study the benefits of the lexical-semantic integration for queries that can potentially benefit from both. We thus leave the evaluation of other queries for future work.

Table 2: The potential improvements in terms of recall of the hybrid approach over the lexical approach. All differences with Lexical are statistically significant.

Method	$c = 500$	$c = 1000$	$c = 1500$	$c = 2000$
Lexical	.429	.538	.596	.635
Semantic	.063	.106	.137	.163
Hybrid	.454	.568	.628	.669
% Improvement	+5.8%	+5.6%	+5.4%	+5.4%

4.1.3 Semantic model implementation. We do not use a pre-trained model; instead, we architected a BERT model using the TensorFlow library with 6 layers, a hidden size of 256, and 4 attention heads, and trained it using the Adam optimizer with a learning rate of 5e-4 and a batch size of 32 for 5 million training steps. We use a vocabulary of 7500 words which was obtained by using a threshold of 300 occurrences of a word in the training set. The semantic retrieval was performed using an approximate in-memory KNN search to enable the efficient parallel execution of the semantic and the lexical retrieval.[3]

4.1.4 Evaluation measures. Since our focus is on improving the recall of retrieval, we report the following evaluation measures: $recall$, Mean Average Precision (MAP), and the total number of relevant documents retrieved for all queries ($\#rel$). Unless stated otherwise, those measures are calculated using the full size of the result list, c ($\in \{500, 1000, 1500, 2000\}$). To measure the robustness of the hybrid approach, we also report the Reliability of Improvement (RI). $RI = \frac{|Q^+|-|Q^-|}{|Q|}$, where $|Q^+|$ and $|Q^-|$ are the number of queries for which the hybrid approach performs better or worse than the lexical baseline, respectively; $|Q|$ is the total number of queries. The two-tailed paired t-test was used to determine statistically significant differences between different methods ($pval < 0.05$).

4.2 Experimental Results

4.2.1 The potential benefits of the hybrid approach. As a first step, we are interested in examining the potential benefit of enriching a lexical-based result list using documents retrieved by a semantic model. Specifically, we are interested to know to what extent the semantic approach can retrieve documents that were not retrieved (or ranked low) by the lexical retrieval model. The results of this analysis can serve as an upper bound for the performance of the hybrid approach. To measure the potential benefits of the hybrid approach, the following experiment was performed. Given two result lists of size c (lexical and semantic), a final result list of size c is generated as well. To do that, we identify relevant documents in the semantic-based result list that do not appear in the lexical-based list. Then, we replace the non-relevant documents in the lexical list with the semantic-based relevant documents.[4] The results of this experiment are reported in Table 2. According to the results, we can see that the lexical approach is much more effective than the semantic approach in terms of recall for all sizes of the result

[2] github.com/castorini/anserini

[3] github.com/google-research/google-research/tree/master/scann

[4] Since we focus on recall, there is no importance for the order of replacement.

Table 3: The performance of the hybrid retrieval approach. All differences in performance (*MAP* and *recall*) between methods in each block are statistically significant.

c	Method	*recall*	*MAP*	*#rel*	*RI*
500	Lexical	.429	.225	11, 585	-
	Hybrid	.441 (+2.8%)	.228	11, 949 (+3.1%)	.413
1000	Lexical	.538	.256	15, 386	-
	Hybrid	.553 (+2.8%)	.259	15, 848 (+3.0%)	.512
1500	Lexical	.596	.269	17, 487	-
	Hybrid	.612 (+2.7%)	.272	18, 033 (+3.1%)	.488
2000	Lexical	.635	.275	18, 997	-
	Hybrid	.653 (+2.8%)	.278	19, 613 (+3.2%)	.446

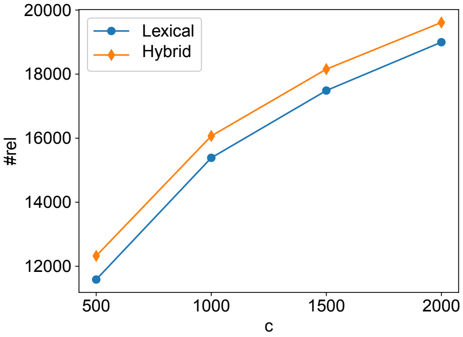


Figure 3: The number of relevant documents when merging a fixed-length semantic-based result list (of 2000 documents) with a lexical-based result list of different lengths.

list. This result shows that the semantic approach cannot replace the classical lexical model in the retrieval stage and explains why previous works only used neural models for the re-ranking stage (e.g., [13, 15]). Yet, this analysis reveals that a semantic model can retrieve a large number of relevant documents that are not included in the lexical-based result list. Specifically, for all sizes of the result list, there is a large and significant improvement in recall when incorporating semantically retrieved results in the lexical list. Furthermore, it is interesting to see that the improvement is stable with respect to the result list size which attests to the potential robustness of the hybrid approach. This result motivates the exploration of automatic approaches for merging the two lists. In the next sections, we show that even when using a simple unsupervised merging approach, significant improvements can be achieved.

4.2.2 Hybrid approach performance. The performance of the hybrid approach is reported in Table 3. The results demonstrate the effectiveness of the hybrid method even when a simple approach is used for the merging stage. Specifically, for all levels of c, the hybrid approach improves over the baseline lexical approach in terms of recall by about 3%. Focusing on the RI measure, we can see that the hybrid approach is also highly robust with respect to the different queries in terms of the recall improvements.

An important question that comes up from the results in Table 3 is: Can the same improvements in recall be achieved by simply considering a longer result list of the lexical model and re-ranking it using RM3? To address this question, the following analysis was performed. Focusing on a semantic-based result list of 2000 documents, we merge it with lexical-based result lists of increasing lengths ($\in \{500, 1500, 1000, 2000\}$), and clip the final result lists to the original length of the lexical result list. The results of this analysis are presented in Figure 3. In the figure, we report the number of relevant documents retrieved for each size of the result list. As can be seen, the number of relevant documents added by the hybrid approach remains stable for all lengths of the lexical list (the value is around 700). This analysis shows that even though we consider longer lexical lists, the semantic approach can still bring the same amount of unique relevant documents on top of it.

4.2.3 Robustness analysis. In this section, we analyze the robustness of the hybrid approach with respect to the different queries. First, we divide the queries in the evaluation set such that the queries in each group have a similar level of increase (or decrease) in recall when using the hybrid retrieval approach, compared to the lexical retrieval baseline; the increase/decrease is measured in percentage; we focus on a result list of 1000 documents. The queries in each group are counted and presented in a histogram in Figure 4. According to the results, it is clear that the hybrid approach is very robust. Specifically, the hybrid approach either improves or does not degrade the performance of the baseline in the majority of cases. According to the results in Figure 4, for 50% of the queries there is an improvement when using the hybrid approach, for 40% there is no change in performance, and for 20% there is a degradation in performance. Yet, while the average percentage of improvement for the good performing queries is around 18%, the performance of the bad performing ones decreases in about 4% only.

In the next analysis, we are interested in examining the performance of different groups of queries, divided based on their performance when using the lexical retrieval model. This analysis can help us better understand the origin of the average overall improvements of the hybrid approach over the lexical model. The results of this analysis are reported in Table 4. To perform the analysis, we split the query set into four equal groups (Q1-4) based on similar performance when using the lexical approach (Q1 are the poorest-performing queries). According to the results, we can see that the improvements of the hybrid approach are much higher for the low quarters with an average improvement of 14% for Q1. In the higher quarter (Q4), on the other hand, there is only a very slight improvement. To further understand the different properties

Table 4: The performance (recall) of four equally sized groups of queries, partitioned based on their performance when the lexical model is used. Statistically significant differences are marked with an asterisk.

Method	Q1	Q2	Q3	Q4
Lexical	.167	.423	.663	.887
Hybrid	.191*	.446*	.674*	.891
% Improvement	+14%	+5.5%	+1.7%	+0.5%

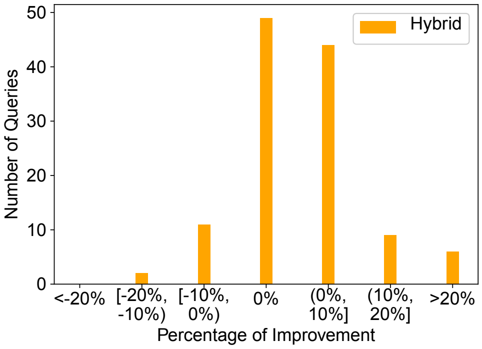


Figure 4: The number of queries in different groups that were divided based on similar level of decrease/increase in performance in the hybrid approach as compared to the lexical retrieval model (in percentage).

of queries in the different performance groups, we perform an analysis of different query properties. Specifically, for each query, the mean, max, and standard deviation of the idf values of its terms is computed; the number of query terms is also calculated. The average values of these measures in each query group are reported in Table 5. According to the results, the mean and max of idf values is higher for the query groups in which the hybrid approach is better performing (for example, comparing the performance of Q1 with that of Q4). A possible explanation for this is that lexical approaches can fail in cases where the query is dominated by a single term that has a high idf value. This might be the case since lexical models often weigh the importance of query terms using a function of idf. An example of such a scenario was also given in Table 1 in the introduction. In that example, we saw that the lexical retrieval model “missed” a relevant document that did not contain a word with a potentially high idf. This observation is further supported by the standard deviation values of the different groups, also reported in Table 5. Finally, the results show that the queries with the better performance when using the hybrid approach are longer. A possible explanation for that is the ability of neural networks to learn semantics using multiple words.

Table 5: Different properties of query groups, partitioned based on their performance when the lexical model was used.

Property	Q1	Q2	Q3	Q4
$Mean(idf)$	10.4	10.4	10.3	9.3
$Max(idf)$	16.9	16.0	16.4	15.2
$Std(idf)$	6.9	6.0	6.5	5.9
Number of terms	3.8	3.9	3.3	3.7

4.2.4 *An analysis of relevant documents.*

In the following, an analysis is performed to shed light on the differences between the relevant documents retrieved by the lexical and the semantic models.

We start the analysis with a case study of three example queries from the query set. These queries were selected since they contain a substantial amount of relevant documents for the two retrieval models, cover diverse topics, and are of different lengths. The first result of this analysis is presented in Table 6. For each query, a semantic and a lexical list of 1000 documents is retrieved. Then, representative terms are extracted from each list using the top k relevant documents in the list, where k is set to be the minimum number of relevant documents between the two lists. The representative terms are then extracted using the $tf.idf$ scoring function. For each query, the number of relevant documents used, $\#docs$, and the Jaccard index (J) between the term lists of the two approaches (of 50 terms) are reported in the header line. Query (a) (“weather related fatalities”) is an example of the case where the semantic terms are related to a narrow topic, while the lexical terms cover a more general topic. Specifically, the semantic list has terms related to the topic of hurricanes (e.g., “hurricane”, “island”, and “mph”), while the lexical terms are all related to the theme of the query, but can hardly be associated with a single topic. In such a case, the hybrid approach can potentially improve over the lexical baseline by strengthening the coverage of a specific aspect of the information need. Query (b) (“automation”) is an example of a case in which the two approaches presumably cover two distinct topics. The semantic terms are quite related to the aspect of computer automation (e.g., “ibm” and “application”), wherein the lexical retrieval model we can see terms related to automation in the traditional industry (e.g., “product” and “staff”). Query (c) (“efforts to enact gun control legislation”) serves as another example for a situation in which the semantic results presumably cover a narrow topic. Specifically, the terms “president” and “bush” might insinuate that. Quantitatively, we can see that the vocabulary of the documents is substantially different for the two models as supported by the low Jaccard index.

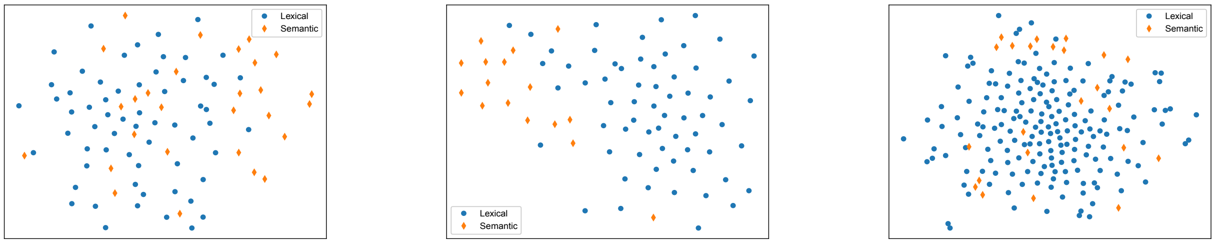
Table 6: Representative terms in relevant documents which were retrieved by the different retrieval models. Boldface: a unique term for a specific model.

(a) *weather related fatalities* ($\#docs$ = 28; J = .333)		(b) *automation* ($\#docs$ = 16; J = .176)		(c) *efforts to enact gun control legislation* ($\#docs$ = 23; J = .282)	
Lexical	Semantic	Lexical	Semantic	Lexical	Semantic
people	storm	**automation**	system	gun	gun
storm	wind	system	**data**	**bill**	**bush**
head	head	**product**	**application**	nra	**text**
weather	**hurricane**	**automate**	software	**control**	weapon
report	people	**operation**	**information**	**drug**	ban
tornado	**mph**	**center**	**new**	law	**say**
wind	weather	**process**	**service**	weapon	**president**
home	**island**	**staff**	**user**	**handgun**	law
today	report	software	**image**	ban	**issue**
service	**inch**	**management**	**ibm**	**wait**	nra

Table 7: The mean and standard deviation of the Jaccard index between the representative terms of the semantic and the lexical retrieval models for different number of terms.

		10	50	100	200
Jaccard	Mean	.184	.169	.162	.156
	Std	.153	.102	.091	.094

The difference between the relevant documents of the two approaches is further emphasized by the visualization presented in



(a) *weather related fatalities*

(b) *automation*

(c) *efforts to enact gun control legislation*

Figure 5: Two-dimensional visualization of the relevant documents in the lexical and the semantic retrieval models.

Figure 5. In the figure, the relevant documents of the two approaches are placed in a two-dimensional space using their $tf.idf$ representations.[5] We focus only on documents that are unique for a specific retrieval model. The vectors were embedded into a two-dimensional space using the t-SNE technique [25]; According to the visualization, it can be seen that the semantic results often form clusters that are located in areas with small (or no) presence of lexical results. In some cases (query (c), for example), the lexical results can form a single dense cluster and the semantic results appear in sparser areas. This analysis shows the potential of the hybrid approach in increasing the diversity and the topic coverage of the result list.

To further support the above findings, a quantitative analysis was performed. For the analysis, all queries with at least five relevant documents, retrieved by each retrieval model, were taken into consideration, resulting in 50 queries. For each query, only the first five relevant documents were used to eliminate any biases regarding the number of documents considered. Then, we examined the average and the standard deviation of the Jaccard index between the term lists of the semantic and the lexical models; this analysis was performed for different numbers of terms. The results, presented in Table 7, show that, in the general case, the overlap between terms in the semantically retrieved documents and the lexically retrieved documents is very low. Moreover, this finding is consistent for different lengths of the term list and is stable over queries as can be attested by the low standard deviation.

The relevant documents in the two approaches can also differ in length as can be seen in Figure 6. To construct the figure, the relevant documents with respect to all queries were pooled, sorted by length, and finally placed in a scatter plot.[6] We can see from the figure that the semantic-based documents are often longer than the lexical-based documents and for about half of these documents the difference can be very large. A possible explanation for this is that classical lexical retrieval models are often designed to penalize long documents in the scoring function. This mechanism, however, does not exist in the semantic-based approaches. Furthermore, it might be the case where semantic approaches can better leverage longer pieces of text and words with low frequencies by using dense representations. Consequently, semantic approaches may be better in retrieving long relevant documents.

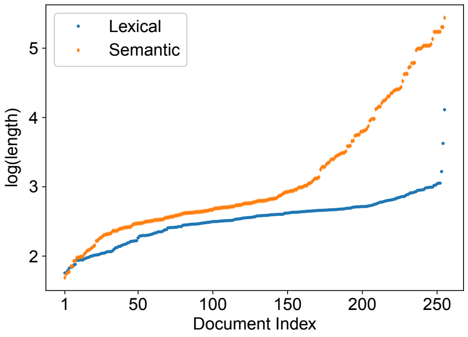


Figure 6: The lengths of relevant documents, retrieved by the semantic and the lexical retrieval models.

5 CONCLUSIONS

Lexical-based retrieval models are the common models used in search engines for the retrieval stage. This work is the first one to systematically study the combination of semantic and lexical models for the retrieval stage of the ad hoc document retrieval task. We proposed a general hybrid approach for document retrieval that leverages both semantic and lexical retrieval models. An in-depth empirical analysis was performed which demonstrated the effectiveness of the hybrid approach and also shed some light on the complementary nature of the lexical and the semantic models.

There are several possible directions for future work that can be tackled. First is the development of more sophisticated approaches for the merging of the lexical and the semantic result lists. Second, in this work we addressed the problem of representing long documents through breaking them into short passages. Instead, more complex representations that take into account document structure can be considered. Finally, it would be interesting to evaluate the effectiveness of the hybrid retrieval approach for other information retrieval tasks including question answering, recommendation systems, and conversational agents.

[5] The vocabulary was restricted to words that appear in at least 10 documents in each document set of a given query.

[6] We used 5 documents per query, resulting in 250 documents overall; note that a point on the x-axis usually refers to two different documents.

REFERENCES

[1] Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. In *Proc. of the 13th Text Retrieval Conference.* 13.

[2] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proc. of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 475–484.

[3] Avinash Atreya and Charles Elkan. 2011. Latent semantic indexing (LSI) fails for TREC collections. *ACM SIGKDD Explorations Newsletter* 12, 2 (2011), 5–10.

[4] Bodo Billerbeck and Justin Zobel. 2004. Techniques for efficient query expansion. In *International Symposium on String Processing and Information Retrieval.* Springer, 30–42.

[5] Bodo Billerbeck and Justin Zobel. 2005. Document expansion versus query expansion for ad-hoc retrieval. In *Proceedings of the 10th Australasian Document Computing Symposium.* Citeseer, 34–41.

[6] Leonid Boytsov, David Novak, Yury Malkov, and Eric Nyberg. 2016. Off the beaten path: Let's replace term-based retrieval with k-nn search. In *Proceedings of the 25th ACM international on conference on information and knowledge management.* 1099–1108.

[7] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems.* 737–744.

[8] William R Caid, Susan T Dumais, and Stephen I Gallant. 1995. Learned vector-space models for document retrieval. *Information processing and Management* 31, 3 (1995), 419–429.

[9] Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)* 44, 1, Article 1 (2012), 50 pages.

[10] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[11] Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. arXiv:1910.10687

[12] Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. arXiv:1901.02860

[13] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proc. of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 65–74.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805

[15] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proc. of the 25th ACM International Conference on Information and Knowledge Management.* 55–64.

[16] Ruiqi Guo, Sanjiv Kumar, Krzysztof Choromanski, and David Simcha. 2016. Quantization based fast inner product search. In *Proc. of the 19th International Conference on Artificial Intelligence and Statistics.* 482–490.

[17] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. [n.d.]. Accelerating Large-Scale Inference with Anisotropic Vector Quantization. ([n. d.]).

[18] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2020. Accelerating Large-Scale Inference with Anisotropic Vector Quantization. In *Proc. of the 37th International Conference on Machine Learning.*

[19] Christophe Van Gysel, Maarten De Rijke, and Evangelos Kanoulas. 2018. Neural vector spaces for unsupervised information retrieval. *ACM Transactions on Information Systems (TOIS)* 36, 4 (2018), 1–25.

[20] Dany Haddad and Joydeep Ghosh. 2019. Learning more from less: Towards strengthening weak supervision for ad-hoc retrieval. In *Proc. of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 857–860.

[21] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using click-through data. In *Proc. of the 22nd ACM International Conference on Information and Knowledge Management.* 2333–2338.

[22] Victor Lavrenko and James Allan. 2006. Real-time query expansion in relevance models. *IR 473, University of Massachusetts Amherst* (2006).

[23] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics.* 6086–6096.

[24] Hao Li, Wei Liu, and Heng Ji. 2014. Two-stage hashing for fast document retrieval. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).* 495–500.

[25] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[26] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web.* 1291–1299.

[27] Bhaskar Mitra, Corby Rosset, David Hawking, Nick Craswell, Fernando Diaz, and Emine Yilmaz. 2019. Incorporating query term independence assumption for efficient retrieval and ranking using deep neural networks. *arXiv preprint arXiv:1907.03693* (2019).

[28] Marius Muja and David G. Lowe. 2014. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 11 (2014), 2227–2240.

[29] Yifan Nie, Alessandro Sordoni, and Jian-Yun Nie. 2018. Multi-level abstraction convolutional model with weak supervision for information retrieval. In *Proc. of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval.* 985–988.

[30] Priyanka Nigam, Yiwei Song, Vijai Mohan, Vihan Lakshman, Weitian Ding, Ankit Shingavi, Choon Hui Teo, Hao Gu, and Bing Yin. 2019. Semantic product search. In *Proc. of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 2876–2885.

[31] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. arXiv:1901.04085

[32] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docTTTTTquery. *Online preprint* (2019).

[33] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. arXiv:1910.14424

[34] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *arXiv preprint arXiv:1904.08375* (2019).

[35] Kezban Dilek Onal, Ye Zhang, Ismail Sengor Altingovde, Md Mustafizur Rahman, Pinar Karagoz, Alex Braylan, Brandon Dang, Heng-Lu Chang, Henna Kim, Quinten McNamara, et al. 2018. Neural information retrieval: At the end of the early years. *Information Retrieval Journal* 21, 2-3 (2018), 111–182.

[36] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proc. of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 232–241.

[37] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proc. of the 23rd ACM International Conference on Information and Knowledge Management.* 101–110.

[38] Tao Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. 2006. Language model information retrieval with document expansion. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference.* 407–414.

[39] Martin Theobald, Ralf Schenkel, and Gerhard Weikum. 2005. Efficient and self-tuning incremental query expansion for top-k query processing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval.* 242–249.

[40] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proc. of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 55–64.

[41] Jinxi Xu and W Bruce Croft. 2017. Query expansion using local and global document analysis. *ACM SIGIR Forum* 51, 2 (2017), 168–175.

[42] Liu Yang, Hamed Zamani, Yongfeng Zhang, Jiafeng Guo, and W Bruce Croft. 2017. Neural matching models for question retrieval and next question prediction in conversation. arXiv:1707.05409

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

[44] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple applications of BERT for ad hoc document retrieval. arXiv:1903.10972

[45] Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing.* 3481–3487.

[46] Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proc. of the 27th ACM International Conference on Information and Knowledge Management.* 497–506.