ExCOIL: Expansion-Aware Contextualized Inverted List for First-Stage Retrieval

Aditya Srikanth Veerubhotla
Language Technologies Institute
Carnegie Mellon University
adityasv@andrew.cmu.edu

Abstract

In this report, we present an important extension to the COIL architecture by enabling term expansion. Our results show that the expansion method we proposed show substantial improvements over COIL and achieves competitive results over different state-of-the-art retrieval models. We perform extensive experiments and analysis to understand the strengths and weaknesses of our system. Our code has been made public for reproducibility and future work [1].

1 Introduction

The recently proposed COIL (Gao et al., 2021) model presents a unique and efficient hybrid approach for performing information retrieval by bridging the gap between Dense Retrieval and Sparse Retrieval. Them model represents a document as a set of contextualized token vectors and performs matching over the common tokens between a query and a document. Their approach combines the strengths of scalable inverted lists for document indexing and soft-semantic matching over contextualized representations for better relevance signals, achieving state-of-the-art performance on the MSMARCO dataset (Nguyen et al., 2016). However, the model still suffers from the vocabulary-mismatch problem, where the query and document use different terms, leading to few/no overlapping terms between the query and a document.

To combat the vocabulary mismatch problem, techniques for performing document/query expansions have been proposed (Abdul-Jaleel et al., 2004; Nogueira et al., 2019b,a) which insert additional tokens to the query/document to reduce the vocabulary mismatch. With the introduction of Deep Learning based Language Models, better techniques for modeling term expansions were developed, resulting in substantial increase in retrieval performance (Formal et al., 2021b; MacAvaney et al., 2020; Zhao et al., 2020). As, these techniques still represent the terms in a query and a document as real values representing their importance (also called *impact* (Mallia et al., 2021)) in the passage, they cannot contain the context in which each of the terms occur, which can be useful for matching.

In this work we propose the ExCOIL architecture, which combines the contextual term representations from COIL with expansions from Deep Learning based Language Models. To the best of our knowledge, this is the first system that combines the scalability of sparse lexical representations, matching power from contextual token representations with trainable expansion mechanism for reducing vocabulary mismatch. Through our experiments we show that our approach achieves state-of-the-art results on the MSMARCO dataset while being efficient in storage and retrieval.

2 Related Work

Classical Information Retrieval systems leverage sparse, lexical, exact match of the tokens between a query and document for matching. This Sparse, Lexical, Exact-Match based Retrieval paradigm, which we call Sparse Retrieval for short has shown great success over time. Sparse Representations allows for the efficient indexing structures such as the Inverted List, on which indexing and retrieval can be performed efficiently. Popular techniques for performing matching using this paradigm are the BM25 model (Robertson and Walker, 1994), which represents the term representations using term occurrence statistics at the local level using the Term Frequency (*tf*) and at the global level using the Inverse Document Frequency (*idf*) and the Statistical

[1] https://github.com/Vaishakh-K/SparseRetrieval/tree/coil-expansion

Language Model (Lafferty and Zhai, 2001), that computes probability of matching by estimating the query likelihood and the document prior. While techniques have been shown to be effective across multiple domains as a strong baseline, they suffer from vocabulary mismatch and their heuristic-driven dataset-agnostic approach have been outperformed with the improvements using supervised Machine Learning based approaches.

Recently, there has been a great interest in the application of Deep Pretrained Neural Language Models for Information Retrieval. Using these language models, there have been approaches for obtaining learned bag-of-words representations which have achieved substantial improvement in terms of retrieval quality. DeepCT (Dai and Callan, 2019), and later the HDCT (Dai and Callan, 2020) models proposed to learn document term representations by regressing over query recall scores. Instead of defining an label score for every term in the document, DeepImpact (Mallia et al., 2021) learns term representations by distantly supervising the term representations using a triplet loss over a scoring function designed on the term representations.

To combat the vocabulary mismatch problem, document expansion techniques such as the Doc2Query model (Nogueira et al., 2019b) and the DocT5Query (Nogueira et al., 2019a) models have been proposed, which generate synthetic queries that are appended to the passage before indexing. In contrast to previous sparse approaches which operated only on the vocabulary of the passage terms, SparTerm (Bai et al., 2020) explicitly performs matching on the entire vocabulary space by creating a vocabulary expansion model called *"Gating Controller"*, and a term importance model called *"Importance Model"*. SPARTA (Zhao et al., 2020) performs matching by cross-matching query terms and document terms and max pooling over document terms before applying a sparsity and non-negative activation for obtaining the matching score. EPIC(MacAvaney et al., 2020) introduces a sparsity inducing activation function coupled with a document level quality scoring function, which scales the relevance up/down depending on the quality of the source. Combining the ideas from the previously mentioned works, SPLADE obtains sparse representations by using a sparsity inducing activation function on top of BERT's (Devlin et al., 2018) MLM logits, and sum-pools over the document terms to obtain a vector in the vocabulary space. SPLADEv2 (Formal et al., 2021a) builds upon SPLADE by changing the pooling from sum to max, and performs Knowledge Distillation (Hinton et al., 2015) from a cross encoder model to obtain state-of-the-art results on the BEIR benchmark (Thakur et al., 2021).

An parallel technique for Information Retrieval is to do away with the token representations in the vocabulary space, and instead focus on learning a dense latent representation to perform matching. This philosophy forms the basis of Dense Retrieval, which projects a query and a document into a latent semantic space where the matching between queries and documents can be performed using a distance measure such as their inner product. The aim of Dense Retrieval is to learn a metric space where the similarity of queries with relevant passages is higher than that of irrelevant passages. With the the introduction of Sentence-Transformers (Reimers and Gurevych, 2019), this field has seen rapid growth. The approaches proposed differ in terms of the language models used (Gao and Callan, 2021), the sampling of negatives (Karpukhin et al., 2020; Xiong et al., 2020; Qu et al., 2020; Lindgren et al., 2021), and the training process (Hofstätter et al., 2021).

Combining different aspects of Dense Retrieval and Sparse Retrieval based approaches, Hybrid systems try to obtain the best of both worlds. A notable example of Hybrid systems is the ColBERT architecture (Khattab and Zaharia, 2020; Santhanam et al., 2021), which uses dense vectors for token representations and performs matching over all the tokens in the query and document using an efficient MaxSim operation. Recently, the COIL architecture (Gao et al., 2021) was introduced which contextualizes the query and document tokens using dense representations, but allows for efficient indexing and retrieval by enforcing lexical matching between query and document tokens. We conduct our experiments primarily on the COIL architecture, and will try to extend this idea by introducing term expansions.

3 Method

In this section, we describe our approach for performing expansion aware contextualized lexical

match. We begin by introducing the COIL model, and describe how its indexing and matching is performed. We then introduce the ExCOIL model for performing matching using expansion aware exact lexical matching.

3.1 Document Matching using COIL

In this section, we describe the matching function used for scoring in COIL (Gao et al., 2021). COIL performs exact lexical matching over the contextual representations of the matching tokens. The representations for query and document tokens are obtained by projecting token representations obtained from a Pretrained Language Model such as BERT (Devlin et al., 2018) into a latent semantic space. Matching of a query with a document is performed by summing up the representations for overlapping query and document terms. The authors present two variants of COIL on the basis of whether the [CLS] token in BERT is used for matching: $COIL_{tok}$ which does not include the classification token and $COIL_{full}$, which includes it. In light of this, the matching function of the variants of COIL can be expressed mathematically as follows: given a query $q = \{q_1, q_2, ..., q_{|q|}\}$ and a document $d = \{d_1, d_2, ..., d_{|d|}\}$:

$$\begin{aligned} H^q &= BERT(q) \\ H^d &= BERT(d) \\ v_i^q &= W_{tok} H_i^q + b_{tok} \\ v_j^d &= W_{tok} H_j^d + b_{tok} \\ s_{match}(v^q, v^d) &= \sum_{q_i \in q \cap d - \{[CLS]\}} \max_{d_j = q_i} (v_i^{q\top} v_j^d) \\ s_{tok}(q, d) &= s_{match}(v^q, v^d) \\ s_{full}(q, d) &= s_{match}(v^q, v^d) + v^{q\top} v^d \end{aligned} \tag{1}$$

where $H_{q/d}$ are the query/document hidden states from the BERT model, W_{tok}, b_{tok} are used to project the hidden representations of a token to a latent semantic space which is then used in the matching function.

It is evident from the above equations that the scoring function critically depends on the matching tokens between a query and a document. While $COIL_{full}$ solves this problem by matching the [CLS] token for each document in the corpus, it is evident that this is not much different from dense retrieval, and maintaining two indexes for dense retrieval and sparse retrieval is not efficient. This motivated us to design a system that ameliorates the vocabulary mismatch problem, while still being sparse in the vocabulary space and resulting in contextualized token representations.

3.2 Expansion aware COIL

Inspired by the insight from SPLADEv2 (Formal et al., 2021a) that Pretrained Language Models learn co-occurrence statistics of terms through the Masked Language Modeling (MLM) pretraining task, and these statistics could be leveraged for performing term expansion, we experiment on how this technique could be incorporated into COIL. As a starting point, we combine the two tasks in a multi-task setup, with the final scoring function being a combination of COIL and SPLADE scoring functions. Subsequently, we try to contextualize the expansion terms. In this section, we describe in detail our approach for obtaining contextualized term expansions.

3.2.1 $COIL_{tok} + CLSExp$: COIL with CLS expansion

As a first step, we hypothesize that term expansion could act as a drop-in replacement for the CLS matching in $COIL_{full}$, making it more efficient by sparsifying the expansions. We use the log-saturation function described in SPLADE (Formal et al., 2021b), henceforth referred to as *log-sat*, for obtaining sparse representations. The scoring technique can be described as follows:

$$\begin{aligned} h_{[CLS]}^q &= BERT(q) \\ h_{[CLS]}^d &= BERT(d) \\ e_{CLS}^q &= f_{log-sat}(W_{MLM} h_{[CLS]}^q) \\ e_{CLS}^d &= f_{log-sat}(W_{MLM} h_{[CLS]}^d) \\ s(v^q, v^d) &= \lambda_{COIL} s_{match}(v^q, v^d) + \lambda_{Exp} {e_{CLS}^q}^\top e_{CLS}^d \end{aligned} \tag{2}$$

where $h^{[CLS]}$ is the hidden representation of the [CLS] token, $e^q \in \mathbb{R}^{|V|}$ is the expansion vector over the vocabulary V, $s_{match}(q, d)$ is the matching function as described in equation (1), and W_{MLM} is the Masked Language Modeling head. The sparsity inducing function $f_{log-sat}$ is defined as:

$$f_{log-sat}(x) = \log(1 + ReLU(x)) \tag{3}$$

which does two things: 1) the ReLU function inside the log ensures that the activations are non-negative, and the adding of one to the output of

ReLU ensures that the activation function is non-negative, and 2) the logarithm ensures that any one term does not dominate the matching.

3.2.2 $COIL_{tok} + TokExp$: COIL with Token expansion

As BERT's MLM objective runs over token-level inputs, it is natural to assume that the true co-occurrence statistics are captured at the token level. To harness this knowledge, we apply the log-sat activation as defined in (3) over the logits of each token, and then pool across the terms in the document terms to obtain a vector in the vocabulary dimension as proposed in SPLADEv2 (Formal et al., 2021a). Formally, this could be expressed as:

$$\begin{aligned} H^q &= BERT(q) \\ H^d &= BERT(d) \\ Z^q &= f_{log-sat}(W_{MLM}H^q) \\ Z^d &= f_{log-sat}(W_{MLM}H^d) \\ e^q_{tok} &= MaxPool_q(Z^q) \\ e^d_{tok} &= MaxPool_d(Z^d) \\ s(q,d) &= \lambda_{COIL}s_{match}(v^q, v^d) + \lambda_{Exp}{e^q_{CLS}}^\top e^d_{CLS} \end{aligned} \tag{4}$$

where $MaxPool_{q/d}(.)$ performs Max Pooling over the terms in the query/document respectively to condense a vector of dimension $|V| \times N$ to $|V|$, where N is the number of tokens in the input to the function.

3.2.3 ExCOIL: COIL with Contextualized Token expansion

As the expansions generated in 3.2.2 are single-dimensional real values representing the impact of the expansion, a natural extension to them would be to contextualize the expansion terms and by grounding them in the query/document. To obtain the expanded document, we follow the following steps: for a query q, and a document d, we first obtain the token representations by projecting the query and the document token representations through to a latent space as follows:

$$\begin{aligned} H^q &= BERT(q) \\ H^d &= BERT(d) \\ v^q &= W_{tok}H^q + b_{tok} \\ v^d &= W_{tok}H^d + b_{tok} \end{aligned} \tag{5}$$

We then use the MLM logit matrix output from BERT and apply $f_{log-sat}$ on this matrix to obtain the expansions, subsequently performing a Max-Pool operation on them to get the expansion vectors. The magnitude of the expansion vectors tell about the importance of the expansion term w.r.t the document. We also store the index in the query where the expansion occurred, and use this to generate the contextualized representation of the expansion term. This can be expressed as follows:

$$\begin{aligned} Z^q &= f_{log-sat}(W_{MLM}H^q) \\ Z^d &= f_{log-sat}(W_{MLM}H^d) \\ e^q &= MaxPool_q(Z^q) \\ e^d &= MaxPool_d(Z^d) \end{aligned} \tag{6}$$

After obtaining the expansion weights and their corresponding indices, we select the top-k expansion weights and perform further operations on them. We perform this operation due to the increased computational complexity, ease of representing the expansions a tensor, and memory constraints on the GPUs. It is important to note that the some of the expansion weights will be zero, and hence will not participate in the output. The context vector for the expansion term is then obtained by multiplying the expansion weight (e^q_i) with the corresponding vector ($v^q_{k^q_i}$). Mathematically, this is expressed as:

$$\begin{aligned} e'^q_{top-k}, ix^q_{top-k} &= SelectTopK(e^q, k) \\ e'^d_{top-k}, ix^d_{top-k} &= SelectTopK(e^d, k) \\ {v_{exp}}^q_i &= e'^q_i \cdot v^q_{k^q_i}, i \in ix^q_{top-k} \\ {v_{exp}}^q_j &= e'^q_j \cdot v^q_{k^q_j}, j \in ix^d_{top-k} \end{aligned} \tag{7}$$

where e'^q, e'^d are the top-k expansion weights for query and document respectively. ix^q, ix^d are the corresponding indices of the query/document words where the top-k activations occur. The scoring of the document is done on the concatenated representations from the inputs with their corresponding expansion terms:

$$\begin{aligned} u^q. &= v^q \oplus v^q_{exp} \\ u^d &= v^d \oplus v^d_{exp} \\ s(q,d) &= s_{match}(u^q, u^d) \end{aligned} \tag{8}$$

Our models were trained using contrastive loss, where the aim is to maximize the likelihood of selecting a relevant document from a collection of documents. Mathematically, this can be expressed as:

$$L_{rank} = \frac{s(q, d^+)}{s(q, d^+) + \sum_{d^- \in \mathbb{D}^-} s(q, d^-)} \quad (9)$$

To control the sparsity of expansions, we used the FLOPS regularizer (Paria et al., 2020), which tries to minimize the expected floating point operations (FLOPs) in matching. The regularizer can be described mathematically as:

$$L_{FLOPS} = \sum_{j \in V} \bar{a}_j^2 = \sum_{j \in V} \left(\frac{1}{N} \sum_{i=1}^{N} w_j^{(d_i)} \right)^2 \quad (10)$$

where a_j is the continuous relaxation of the activation of a term in the vocabulary, which is estimated as the probability of activation, approximated by computing the activation of the term in a batch of size N.

The overall loss function is then defined as:

$$L = L_{rank} + \lambda_q L^q_{FLOPS} + \lambda_d L^d_{FLOPS} \quad (11)$$

4 Experiments

In this section, we describe the experiments that were performed to show the effectiveness of our technique. Our experiments were conducted on the MSMARCO dataset (Nguyen et al., 2016), on the passage ranking task in the full ranking setting. This dataset contains 8.8M passages, with an average length of around 60 tokens, and approximately 1.1 judged passages per query. In line with other approaches, we report the MRR@10 and Recall@1k metrics.

We initialized our models with DistilBERT (Sanh et al., 2019) owing to the smaller footprint of the model while obtaining similar performance. While training, used a learning rate of 5e-6 with a linear schedule and a warmup ratio of 0.1 and set a batch size of 8. For the negatives, we used a combination of negatives sampled from BM25, which were 16 in count and in-batch negatives, leading to 127 negatives per example. We set the token dimension to be 32, $\lambda_q = 0.0008$, $\lambda_d = 0.0006$, similar to the setting reported in SPLADEv2 (Formal et al., 2021a). Training of our model was performed for 5 epochs.

For $COIL_{tok} + CLSExp$ and $COIL_{tok} + TokExp$, we set the expansion weights and token weights to be equal, i.e. $\lambda_{COIL} = \lambda_{Exp} = 1$, following our analysis from ablations which is mentioned in Table 2 and is discussed later. In line with the literature that showed that better negatives are crucial for better performance on the retrieval task, we use the hard negatives mined by the COIL authors[2]. We then analyse the effect of initializing our best token expansion model and performing contextualized expansion, for this we use the $COIL_{tok} + TokExp$ + hard negatives, which we will call ExCOIL-PT. Our final model ExCOIL, is the combination of token-level expansions obtained from the MLM logits that are contextualized using the token representations from the document and trained using hard negatives mined mentioned previously. Owing to the computational complexity of evaluation, we evaluate our models at the end of training, similar to (Gao et al., 2021). Our code was based on the COIL repository [3]

5 Results and Discussion

Table 1 shows the results of our experiments. We compare against state-of-the-art dense retrieval techniques such as ANCE (Xiong et al., 2020), TAS-B (Hofstätter et al., 2021), RocketQA (Qu et al., 2020) and coCondenser (Gao and Callan, 2021). It can be observed that ExCOIL obtains competitive performance to these dense retrieval models. The higher performance of dense retrieval methods can be attributed to the fact that additional training techniques were introduced, from better hard negative mining to knowledge distillation (Hinton et al., 2015) from cross encoders, which led to substantial improvement in retrieval performance. These techniques are used to better train the retriever by reducing the noise in the gradients caused by poor negatives. An extension for this line of work is to train our system on the negatives mined by these approaches.

In the case of Sparse Retrieval as well, our method shows strong performance, being only outperformed by DistilSPLADE-max, which relied on the pseudo-labels of a Cross-Encoder for better training signals. This is not sur-

[2] https://github.com/luyug/COIL#msmarco-passage-ranking

[3] https://github.com/luyug/COIL

Model	MRR@10	Recall@1k
*Dense Retrieval**		
ANCE (Xiong et al., 2020)	0.312	0.941
TAS-B (Hofstätter et al., 2021)	0.347	0.978
RocketQA (Qu et al., 2020)	0.370	0.979
coCondenser (Gao and Callan, 2021)	0.382	0.984
*Sparse Retrieval**		
BM25	0.184	0.853
DeepCT (Dai and Callan, 2019)	0.243	0.913
DocT5Query (Nogueira et al., 2019a)	0.277	0.947
SparTerm (Bai et al., 2020)	0.279	0.925
DeepImpact (Mallia et al., 2021)	0.326	0.948
SPLADE (Formal et al., 2021b)	0.322	0.955
SPLADE-max (Formal et al., 2021a)	0.340	0.965
DistilSPLADE-max (Formal et al., 2021a)	0.368	0.979
*Hybrid approaches**		
$COIL_{tok}$ (Gao et al., 2021)	0.341	0.949
COIL-full (Gao et al., 2021)	0.355	0.963
ColBERT (Khattab and Zaharia, 2020)	0.360	0.968
ColBERTv2 (Santhanam et al., 2021)	0.397	0.984
Our experiments		
$COIL_{tok}$ (repro)	0.345	0.945
COIL-full (repro)	0.352	0.963
$COIL_{tok} + CLSExp$	0.346	0.947
$COIL_{tok}$ + DocT5Query	0.358	0.966
$COIL_{tok} + TokExp$	0.357	0.963
$COIL_{tok}$ + HN	0.345	0.945
$COIL_{tok} + TokExp$ + HN	0.370	0.968
ExCOIL-PT	0.210	0.826
ExCOIL	0.364	0.971

Table 1: Results of our experiments on MSMARCO dev set. The results mentioned in * are copied from the corresponding works.

prising, as the baseline COIL model shows substantial improvement over other Sparse Retrieval techniques. However, the addition of the token-expansion system improved the MRR and Recall number even further, showing that there is a non-trivial improvement shown by this technique.

When compared with hybrid approaches, the better performance of ColBERT approaches could be explained by the cross-matching over every query term with every document term. This allows for the model to bypass the vocabulary mismatch problem, and obtain better representations. COIL however, performs matching only on the common terms, and hence suffers from the vocabulary mismatch problem. The addition of DocT5Query ameliorates this problem by generating synthetic queries. However, the increased performance is obtained only after sampling many queries, which are 80 in this case, with each of them being around 10 tokens long. This causes the index to blow up. Our technique adds on an average of 16 tokens in the query increasing its length from an average of 7 tokens to 23 tokens, and 180 tokens in the document which goes up from 71 tokens to 251 tokens, reducing the storage footprint while retaining the performance. An extension of this work is to reduce the number of work by investigating other sparsity regularization techniques. Finally, we observe that $COIL_{tok} + CLSExp$ performs worse than $COIL_{tok} + TokExp$, which can be attributed to the fact that the CLS token was never trained for obtaining expansion tokens, while the MLM logits act as a good solution for expansion and matching.

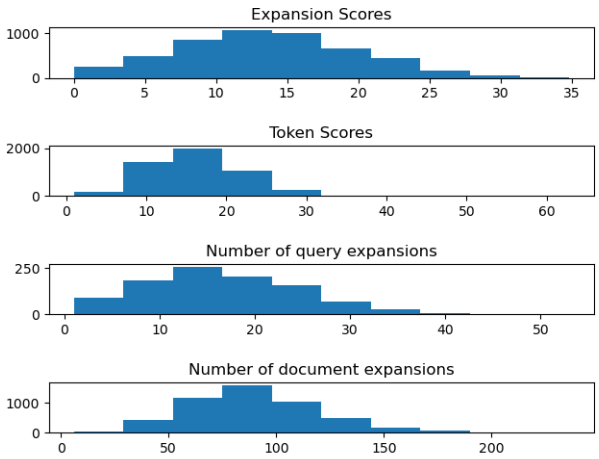

Figure 1: Score and Expansion distribution of the ExCOIL model

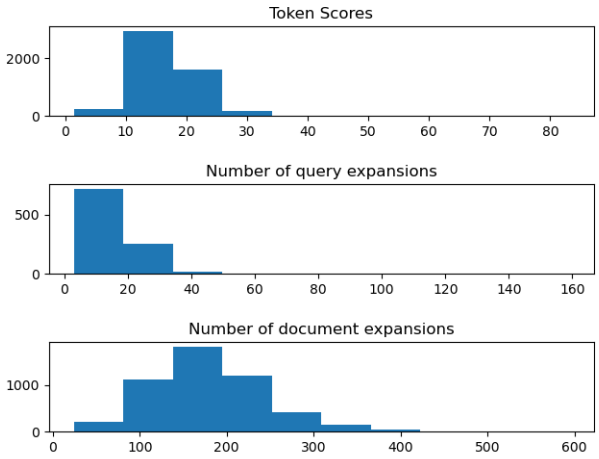

Figure 2: Score and Expansion distribution of the ExCOIL model

5.1 Analysis of the Expansions

Figure 1 shows the expansion score distribution. It is evident that the score distribution is obtained from the COIL matching is similar to that of the expansion matching, which suggests that both components are equally important for obtaining good performance. To confirm this, we conducted an ensemble analysis of the scores from the rankings obtained from the COIL and SPLADEv2 models, which is presented in Table 2 which shows that the best performance occurs when both the distributions are given equal weight.

Similarly, Figure 2 shows the score distribution for the ExCOIL model, and interestingly, this model also has a similar mean for scores, located between 10-20. The number of query expansions also seem to be similar, with a mean at around 16. There is a drastic shift in the number of document, with a near doubling of the number of expansions generated. Better tuning of the hyperparameters could potentially reduce the number of expansions, and we leave this to future work. An example of the document and query expansion are presented in Table 3. It can be seen that the expansions are relevant, and there are essential expansions, from *il* in query to *Illinois* in the document, which leads to the matching of the query with the document terms. This example was chosen to illustrate the impact of document expansion, as $COIL_{tok}$ was unable to retrieve the document, but with expansions, the document was present in the retrieved set. Table 4 clearly shows the improvement of our expansion model, which increased the number of matching tokens from 4.6 to 11.9 by almost 2.5 times, leading to the model giving better matching scores.

5.2 Failure Cases

An interesting finding that was observed was the overwhelming number cases where a relevant document was retrieved, but as it did not match the labelled document, the model was penalized unnecessarily. In our analysis, of a random sample of 50 queries, we observed this "false negative" problem at 46 of the instances. This is due to the sparse judgement performed of MSMARCO dataset. An extensive analysis of this model on diverse datasets would show the true strengths and weaknesses of our approach and we leave this for future work.

Of the places where the model did fail, the failure cases can be categorized into two classes: 1. Matching issues 2. Expansion issues. Matching issues are where the model outputs score which is not aligned with the relevance of the query-document pair. Expansion issues pertain to incorrect expansion tokens produced by the model.

An example of matching issue can be observed in Table 5, where the ExCOIL model matches a majority of keywords in the query, but fails to understand the fact that the query is looking for how pyuria is caused by UTIs. This forms the major bucket of errors that we observed, and we believe that with the improvement in representation learning, these would be fixed, and leve it to future work. For expansion errors, a frequent pattern that was observed was that the model to be unable to understand acronyms. The model neither able to resolve the acronym into its constituent words, nor

μ	MRR@10	Recall@1K
0	0.3381	0.9566
0.1	0.3386	0.9566
0.2	0.3471	0.9566
0.3	0.3475	0.9572
0.4	**0.3485**	0.9613
0.5	0.3471	**0.9619**
0.6	0.3453	0.9555
0.7	0.3416	0.9466
0.8	0.3366	0.9434
0.9	0.3332	0.9424
1	0.3284	0.9422

Table 2: Results from ensembling SPLADE and COIL scores. Interpolation was performed by the following formula: $s(q,d) = \mu s_{COIL_{tok}}(q,d) + (1-\mu)s_{SPLADE}(q,d)$

Query QId: 1004228
when is champaign il midterm elections
Query Expansions
('date', '1.93'), ('mid', '1.42'), ('election', '1.37'), ('il', '1.34'), ('elections', '1.10'), ('champaign', '1.10'), ('day', '0.59'), ('vote', '0.38'), ('session', '0.34'), ('political', '0.23'), ('time', '0.09'), ('illinois', '0.03')
Document DocId: 7256085
Illinois elections, 2018. A general election will be held in the U.S. state of Illinois on November 6, 2018. All of Illinois' executive officers will be up for election as well as all of Illinois' eighteen seats in the United States House of Representatives.
Document Expansions
('date', '1.98'), ('illinois', '1.80'), ('election', '1.77'), ('elections', '1.76'), ('2018', '1.72'), ('il', '1.66'), ('executive', '1.58'), ('general', '1.56'), ('november', '1.53'), ('state', '1.40'), ('house', '1.38'), ('senate', '1.34'), ('representative', '1.28'), ('representatives', '1.25'), ('officers', '1.24'), ('elected', '1.23'), ('day', '1.18'), ('assembly', '1.11'), ('vote', '1.04'), ('will', '1.01'), ('congressional', '1.00'), ('us', '0.99'), ('political', '0.99'), ('officer', '0.98'), ('politics', '0.95'), ('legislative', '0.89'), ('congress', '0.84'), ('ag', '0.84'), ('senator', '0.82'), ('elect', '0.79'), ('seat', '0.78'), ('deadline', '0.76'), ('be', '0.72'), ('statewide', '0.72'), ('arc', '0.69'), ('democracy', '0.68'), ('delegate', '0.67'), ('referendum', '0.64'), ('u', '0.63'), ('legislature', '0.60'), ('sunday', '0.59'), ('bet', '0.59'), ('chicago', '0.58'), ('democratic', '0.57'), ('dates', '0.57'), ('sa', '0.57'), ('electoral', '0.56'), ('held', '0.55'), ('ia', '0.55'), ('number', '0.54')
Intersection terms
'time', 'il', 'date', 'election', 'day', 'illinois', 'political', 'elections', 'vote'

Table 3: Example of query and document expansion. The example was randomly picked from the set of documents where the ExCOIL was able to retrieve the correct document and COIL-tok was not. For brevity, the top-50 expansions and their scores are displayed.

Model	Average number of intersecting tokens
COIL	4.634
ExCOIL	11.937

Table 4: Number of tokens that are intersecting between a query and a document. Higher number of intersection terms would ensure better matching.

Query QId: 183723
explain how urinary tract infections cause both hematuria and pyuria
Labelled Document DocId: 7360146
Pyuria is the presence of white blood cells (leukocytes) in the urine (6 to 10 or more neutrophils per high power field of unspun, voided mid-stream urine). Pyuria is not a diagnosis; it is a laboratory finding in many diseases, most commonly urinary tract infections (UTI). Pyuria usually indicates that bacteria have invaded the upper or lower urinary tract, invoking an inflammatory response of the lining of the urinary tract (urothelium) in that location.
Retrieved Document DocId: 8554876
There are many possible causes of hematuria, including: Urinary tract infection Hematuria can be caused by an infection in any part of the urinary tract, most commonly the bladder (cystitis) or the kidney (pyelonephritis). Kidney stones . Tumors in the kidney or bladder

Table 5: Example of matching failure

Query QId: 1101845
wnli phone number
Query Expansions
., rec, word
Retrieved Document DocId: 7144526
This Cenlar phone number is ranked #2 out of 4 because 18,246 Cenlar customers tried our tools and information and gave us feedback after they called. The reason customers call 877-680-5583 is to reach the Cenlar Customer Service department for problems like Request a loan, Eligibility question, Repayment question, Overcharge/Strange charge, Extension.
Intersection terms after expansion
'telephone', '##nl', 'number', 'numbers', 'mill', '##tl', 'phone'
Ground Truth Document
Contact WhyNotLeaseIt Customer Service, available seven days a week to assist you. They can be reached by phone at 1-855-965-4669 or by email at sears@whynotleaseit.com. You can also access information regarding your lease at the online Customer Service Center. Not a member?
Intersection terms after expansion
'phone', 'numbers', 'telephone', 'number'

Table 6: Example of query and document expansion failure

is able to construct an acronym out of a candidate noun phrase. An illustration of this problem can be viewed in Table 6. While understanding when an noun phrase becomes an acronym is hard, automatically, we believe that the integration of a knowledge base to the model would help this issue and leave it for future work.

6 Conclusion

In this work, we presented an important extension to COIL architecture, where we introduced document expansion to ameliorate vocabulary mismatch problem faced by lexical retrievers. The results we obtained show that there is a substantial improvement in performance obtained due to expansion, and this system achieves strong results in comparison to multiple state-of-the art baselines. Our analysis showed that this is an efficient solution for obtaining contextualized term representations that are effective at performing matching. We also highlighted to our models shortcomings in terms of their ability to match beyond keyword matching and handling named entities, particularly acronyms. With our analysis, we hope that new research directions arise in terms of obtaining contextualized expansions.

Acknowledgements

We would like to thank Professor Jamie Callan, Zhen Fan and Vaishakh Keshava for their valuable

contributions to this work. Without their feedback, this work would not have come to this stage. We would like to thank Tanmay Kulkarni, for his feedback and help with structuring the document.

References

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. *Computer Science Department Faculty Publication Series*, page 189.

Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. Sparterm: Learning term-based sparse representation for fast text retrieval. *CoRR*, abs/2010.00768.

Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. *CoRR*, abs/1910.10687.

Zhuyun Dai and Jamie Callan. 2020. *Context-Aware Document Term Weighting for Ad-Hoc Search*, page 1897–1907. Association for Computing Machinery, New York, NY, USA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021a. SPLADE v2: Sparse lexical and expansion model for information retrieval. *CoRR*, abs/2109.10086.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021b. SPLADE: sparse lexical and expansion model for first stage ranking. *CoRR*, abs/2107.05720.

Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval. *CoRR*, abs/2108.05540.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. COIL: revisit exact lexical match in information retrieval with contextualized inverted list. *CoRR*, abs/2104.07186.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. *CoRR*, abs/2104.06967.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *CoRR*, abs/2004.04906.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. *CoRR*, abs/2004.12832.

John Lafferty and Chengxiang Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 111–119, New York, NY, USA. Association for Computing Machinery.

Erik Lindgren, Sashank Reddi, Ruiqi Guo, and Sanjiv Kumar. 2021. Efficient training of retrieval models using negative cache. In *Advances in Neural Information Processing Systems*, volume 34, pages 4134–4146. Curran Associates, Inc.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Expansion via prediction of importance with contextualization. *CoRR*, abs/2004.14245.

Antonio Mallia, Omar Khattab, Nicola Tonellotto, and Torsten Suel. 2021. Learning passage impacts for inverted indexes. *CoRR*, abs/2104.12016.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019a. From doc2query to doctttttquery. *Online preprint*, 6.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019b. Document expansion by query prediction. *CoRR*, abs/1904.08375.

Biswajit Paria, Chih-Kuan Yeh, Ian En-Hsu Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. 2020. Minimizing flops to learn efficient sparse representations. *CoRR*, abs/2004.05665.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. *CoRR*, abs/2010.08191.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94*, pages 232–241. Springer.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *CoRR*, abs/2112.01488.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *CoRR*, abs/2104.08663.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *CoRR*, abs/2007.00808.

Tiancheng Zhao, Xiaopeng Lu, and Kyusong Lee. 2020. SPARTA: efficient open-domain question answering via sparse transformer matching retrieval. *CoRR*, abs/2009.13013.