Unsupervised Document Expansion for Information Retrieval with Stochastic Text Generation

Soyeong Jeong[1] **Jinheon Baek**[2] **ChaeHun Park**[1] **Jong C. Park**[1*]
School of Computing[1] Graduate School of AI[2]
Korea Advanced Institute of Science and Technology[1,2]
{syjeong,ddehun,park}@nlp.kaist.ac.kr
jinheon.baek@kaist.ac.kr

Abstract

One of the challenges in information retrieval (IR) is the *vocabulary mismatch* problem, which happens when the terms between queries and documents are lexically different but semantically similar. While recent work has proposed to expand the queries or documents by enriching their representations with additional relevant terms to address this challenge, they usually require a large volume of query-document pairs to train an expansion model. In this paper, we propose an **U**nsupervised **D**ocument **E**xpansion with **G**eneration (UDEG) framework with a pre-trained language model, which generates diverse supplementary sentences for the original document without using labels on query-document pairs for training. For generating sentences, we further stochastically perturb their embeddings to generate more diverse sentences for document expansion. We validate our framework on two standard IR benchmark datasets. The results show that our framework significantly outperforms relevant expansion baselines for IR.

1 Introduction

Information retrieval (IR) is the task of retrieving the most relevant documents, including scientific ones (Boudin et al., 2020; Noh and Kavuluru, 2020), for a given query. IR systems have received considerable attention as they are not only required to search documents for information, but are also used as a core component in various downstream language understanding tasks such as open-domain question answering (Seo et al., 2019; Qu et al., 2020), fact verification (Thorne et al., 2018; Li et al., 2020) and information extraction (Narasimhan et al., 2016; Das et al., 2020).

As the simplest approach to IR tasks, classical term-based ranking models, such as BM25 (Robertson et al., 1994) and Query Likelihood (QL) models (Zhai and Lafferty, 2017), have been widely used. These term-based ranking models measure the lexical overlaps between query and document pairs using a sparse representation of words, to match the relevant documents for the given query. Notwithstanding their simplicity, they achieve decent performances, even compared to the recent dense representation models (Lin, 2019; Xiong et al., 2020), which require a large number of paired query-document samples. However, these term-based sparse models are intrinsically vulnerable to the *vocabulary mismatch* problem, which happens when a query and its relevant document are lexically divergent.

Thus, we should address the limitations of both sparse and dense models, about the vocabulary mismatch problem and the need for a large amount of training data, respectively. Along this line, there are methods that expand queries and documents with their relevant terms. They include document expansion methods (Nogueira et al., 2019; Boudin et al., 2020) that introduce additional context-related terms to given documents and query expansion methods (Mao et al., 2020; Claveau, 2020) that augment given queries with additional terms. By doing so, we can explicitly generate lexically richer documents or queries.

Compared with query expansion, document expansion has two strengths. First, a document expansion model can generate much more relevant terms for the given document, since documents are generally much longer than queries. Also, documents can be expanded during indexing time so that the responding process for the user's query is not delayed, in contrast to queries that must be expanded during retrieval time. Thus, document expansion is more appropriate for a real-time system, together with making available more context-related words from the given information (Nogueira et al., 2019).

In this work, we focus on document expansion, and propose to abstractly generate the key infor-

* Corresponding author

Proceedings of the Second Workshop on Scholarly Document Processing, pages 7–17

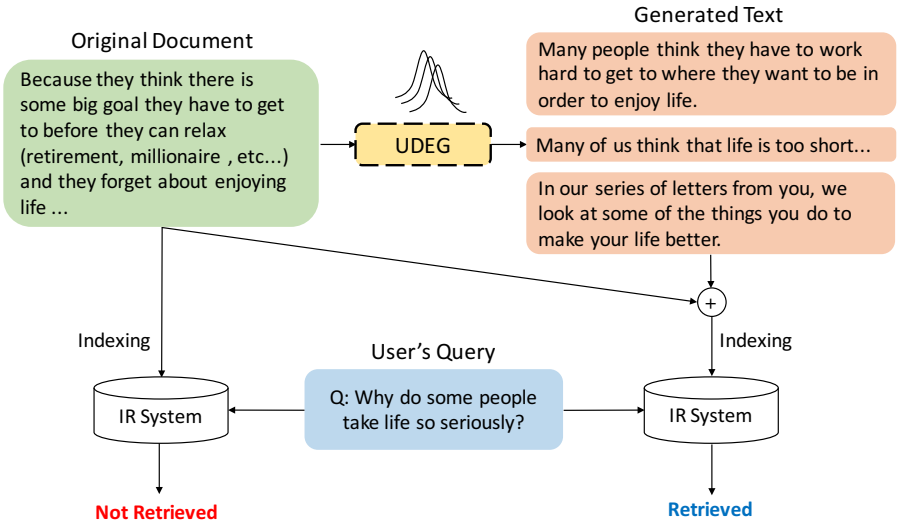

Figure 1: The overall framework of our Unsupervised Document Expansion with Generation (UDEG), where the example is generated from our framework. Given an original document (green box), our UDEG framework stochastically generates several sentences (orange box) relevant to the given document, and augments the generated sentences to the input document to improve its expressiveness. After every document in the corpus is expanded, documents are indexed in the IR system, and searched in response to the given query.

mation corresponding to the given document in an unsupervised manner, henceforth referred to as *Unsupervised Document Expansion with Generation* (UDEG). We first generate document-related sentences using a pre-trained language model, and then stack up the newly generated sentences on the original documents to enrich the expressiveness of document representation. Specifically, in order to generate sentences containing particular information for the documents, we use a language model that is already trained for summarizing sentences from a sufficient amount of texts. However, such a scheme generates only one static sentence at a time, so we further propose to stochastically generate multiple relevant sentences for the given document. This helps the proposed UDEG framework to minimize the vocabulary mismatch cases by generating many relevant words, which reflect diverse points of view for the given document. The overall UDEG framework is illustrated in Figure 1.

We experimentally validate the proposed UDEG framework on standard benchmark datasets for IR tasks, ANTIQUE (Hashemi et al., 2020) and MS MARCO (Nguyen et al., 2016), with five different evaluation metrics. The experimental results show that our framework outperforms all baselines on all evaluation metrics by a large margin. Also, a detailed analysis of UDEG shows that its stochastic generation significantly improves the IR performances, and that our UDEG framework does not depend on specific language models for generation.

Our contributions in this work are threefold:

- To mitigate the vocabulary mismatch problem, we present a novel document expansion framework that augments the document with abstractly generated sentences without using paired query-document data for training.
- Under an unsupervised document expansion framework, we generate document-related sentences with a pre-trained language model, and further stochastically perturb the embeddings for more diverse sentences.
- We show that our framework achieves outstanding performances on benchmark datasets for IR tasks with various evaluation metrics.

2 Related work

Information Retrieval A two-stage pipeline is the most prominent approach for IR. This pipeline first retrieves query-relevant documents with their sparse representations, and then re-ranks them by using neural networks (Mitra and Craswell, 2018; Nogueira et al., 2019, 2020). In this two-stage pipeline, the overall performance is critically dependent on the first retrieval stage, since the failure of the retrieval stage would highly affect the second re-ranking stage. Therefore, this bottleneck on the first stage has to be addressed for performance enhancement (Karpukhin et al., 2020). BM25 and

query likelihood (QL) are the most popular *ad-hoc* retrieval models for the first stage (Nogueira et al., 2019; Boudin et al., 2020; Tang and Arnold, 2020). More recently, instead of using sparse models, methods of using dense representations have been proposed (Karpukhin et al., 2020; Xiong et al., 2020; Qu et al., 2020), which can help alleviate the vocabulary mismatch problem through a dense representation space. However, recent work has revealed the limitations on their performance and efficiency (Lin, 2019; Xiong et al., 2020; Luan et al., 2020). Furthermore, these dense representation methods are based on supervised learning, where pairs of query and related-document are usually required to ensure reasonable performance.

Query / Document Expansion Query and document expansions have been widely used in IR systems. In terms of query expansion, Jaleel et al. (2004) proposed pseudo relevance feedback (RM3), which is revisited in more recent work (Dibia, 2020; Mao et al., 2020) for its strength. There are also methods that expand queries using generation schemes (Mao et al., 2020; Claveau, 2020). However, query expansion suffers from its intrinsic drawbacks, as queries need to be manipulated during the retrieval phase and have relatively less information than documents (Nogueira et al., 2019). Thus, we take the alternative route: expanding documents. Nogueira et al. (2019) and Tang and Arnold (2020) proposed to expand documents with generated text using a supervised model trained on query-document pairs. In contrast, our framework generates document-related sentences regardless of the existence of the corresponding query. Boudin et al. (2020) proposed to expand documents with sequence-to-sequence models which output keyphrases; however, their models have to be trained from scratch on a specific domain.

Document-relevant Text Generation In order to enrich the given document efficiently with the document-relevant text, such text should contain the document's key context which can appear in the summarized sentence. Earlier, Erkan and Radev (2004) and Mihalcea and Tarau (2004) proposed unsupervised models of extracting key sentences, which are adopted in various recent work for their robustness (Nikolov and Hahnloser, 2020a; Zhang et al., 2020b; Kazemi et al., 2020). In contrast, an abstractive approach aims at generating summarized sentences containing novel terms that might not exist in the given document (Zhang et al., 2020a; Yang et al., 2020). Nikolov and Hahnloser (2020b) proposed to first extract the key sentences and then paraphrase them with back-translation. Recent work has reported that the improved performance of text summarization approaches is attributed to the pre-trained language models (Zhang et al., 2020a; Lewis et al., 2020; Xu et al., 2020). In this work, we aim at abstractly generating a document-related sentence with a pre-trained language model and further propose to diversely generate sentences with stochastic perturbation, not just using a single summarized sentence.

3 Method

Our goal is to expand the document for IR tasks by generating document-related text, which contains novel but semantically similar terms for the given document without using query-document pairs. In this section, we describe formal description of the IR task.

3.1 Preliminaries

We begin with a formal description of the IR task, and then introduce a document expansion scheme.

Information Retrieval The objective of an IR task is to retrieve the most relevant document $\boldsymbol{d} \in \mathcal{D}$ for the given query $\boldsymbol{q} \in \mathcal{Q}$, where $\mathcal{Q}$ and $\mathcal{D}$ indicate query and document set, respectively. Note that the query and document pair can be represented as either sparse (Robertson et al., 1994; Zhai and Lafferty, 2017) or dense (Lin, 2019; Xiong et al., 2020), which gives rise to different implementation details.

Suppose that we are given a query-document pair $(\boldsymbol{q}, \boldsymbol{d})$ in the correct query-document set τ: $(\boldsymbol{q}, \boldsymbol{d}) \in \tau$, where $\tau \subset \mathcal{Q} \times \mathcal{D}$. Then, the system should retrieve the most relevant document $\boldsymbol{d}$ for the given query $\boldsymbol{q}$ in the correct query-document set τ, denoted as follows:

$$\max \sum_{(\boldsymbol{q},\boldsymbol{d})\in\tau} f(\boldsymbol{q}, \boldsymbol{d}), \tag{1}$$

where $f : \mathcal{Q} \times \mathcal{D} \rightarrow \mathcal{R}$ is a score function that measures the similarity of the correct query-document pairs, to retrieve the most relevant document for the given but unseen query at test time.

Document Expansion While an IR system can work alone as in Equation 1 by using either sparse or dense representations for queries and documents,

Document (Input): Because they think there is some big goal they have to get to before they can relax (retirement, millionaire, etc...) and they forget about enjoying life ...
Ext (Output): relax (retirement, millionaire , etc...)
Abs (Output): Many people think they have to work hard to get to where they want to be in order to enjoy life.
Abs + S (Output): 1) Many people think they have to work hard to get to where they want to be in order to enjoy life. **2)** Many of us think that life is too short... **3)** In our series of letters from you, we look at some of the things you do to make your life better.

Table 1: Examples of the generated text from different text-generation schemes. Generated terms are appended to the input document before indexing them for the IR system. Ext and Abs denote the extractive and abstractive generation, respectively. Also, + S symbol denotes the stochastic generation.

we need to deal with the vocabulary mismatch problem, which happens when the terms between queries and documents are lexically different but semantically related. To address this problem, we focus on the document expansion scheme, which augments the document with relevant terms to make a richer document.

Formally, the goal of document expansion is to generate semantically relevant terms $\boldsymbol{t} = [t_i]_{i=1}^{K}$ for the given document $\boldsymbol{d} \in \mathcal{D}$, denoted as follows:

$$[t_i]_{i=1}^{K} = g(\boldsymbol{d}; \boldsymbol{\theta}), \tag{2}$$

where K is the number of terms associated with each document $\boldsymbol{d}$ and g is the document expansion model parameterized by $\boldsymbol{\theta}$. After generating relevant terms $\boldsymbol{t} = [t_i]_{i=1}^{K}$ for the document, we concatenate them with the original document $\boldsymbol{d}$ to construct the more meaningful document-representation $\bar{\boldsymbol{d}}$, denoted as follows:

$$\bar{\boldsymbol{d}} = [\boldsymbol{t} \oplus \boldsymbol{d}], \tag{3}$$

where $\oplus$ is the concatenation operation.

By expanding relevant terms to the given document with the generation model g, the similarity between the query $\boldsymbol{q}$ and expanded document $\bar{\boldsymbol{d}}$ becomes stronger than the similarity between query $\boldsymbol{q}$ and original document $\boldsymbol{d}$, as follows: $f(\boldsymbol{q}, \boldsymbol{d}) \leq f(\boldsymbol{q}, \bar{\boldsymbol{d}})$. In order to maximize the similarity score between $\boldsymbol{q}$ and $\bar{\boldsymbol{d}}$, we need the model g that generates document-related terms without using labels of query-document pairs τ for training, which we describe in the next subsection.

3.2 Unsupervised Text Generation for Document Expansion

We now describe our *Unsupervised Document Expansion with Generation* (UDEG) framework, which generates relevant terms for the given document $\boldsymbol{d}$ without using labels on query-document pairs $(\boldsymbol{q}, \boldsymbol{d}) \in \tau$. We first introduce the extractive and abstractive text generation schemes, which are two representative methods for unsupervised text generation, and then propose a stochastic generation scheme for a richer vocabulary.

Extractive Text Generation Extractive text generation is to select the representative words or sentences on the given document. Formally, an extractive text generation scheme is defined as follows:

$$\begin{aligned} \boldsymbol{t}_{ext} = [(t_{ext})_i]_{i=1}^{K} = g_{ext}(\boldsymbol{d}; \boldsymbol{\theta}_{ext}), \\ [(t_{ext})_i]_{i=1}^{K} \subset \boldsymbol{d}, \end{aligned} \tag{4}$$

where g_{ext} is an extractive text generation model parameterized by $\boldsymbol{\theta}_{ext}$. After extracting terms $\boldsymbol{t}_{ext} = [(t_{ext})_i]$, they are used to expand the document as in Equation 3 (i.e., $\bar{\boldsymbol{d}} = [\boldsymbol{t}_{ext} \oplus \boldsymbol{d}]$), which can enrich the representation of the given document by counting important terms multiple times (See Table 1 for examples of extractive generation).

Abstractive Text Generation While the previously described extractive text generation model aims at enriching the given document with key terms extracted from it, the expressiveness of this extractive scheme is highly restricted since novel but semantically similar terms cannot be generated as in Equation 4: $[(t_{ext})_i]_{i=1}^{K} \subset \boldsymbol{d}$. To overcome this limitation, one should further consider generating related-terms that are not contained in the original document. To this end, we propose an abstractive text generation model to obtain the relevant but novel terms for the given document $\boldsymbol{d}$.

Formally, novel terms for the original document are denoted as $[(t'_{abs})_l]_{l=1}^{N} \not\subset \boldsymbol{d}$, whereas existing terms on the document are denoted as $[(t_{abs})_j]_{j=1}^{K-N} \subset \boldsymbol{d}$. N is the number of newly generated document-related terms. Then, an abstractive generation model is defined as follows:

$$\begin{aligned} \boldsymbol{t}_{abs} = \left[[(t'_{abs})_l]_{l=1}^{N} \oplus [(t_{abs})_j]_{j=1}^{K-N} \right] \\ = g_{abs}(\boldsymbol{d}; \boldsymbol{\theta}_{abs}), \end{aligned} \tag{5}$$

where g_{abs} is the abstractive generation model parameterized by $\boldsymbol{\theta}_{abs}$. We provide concrete examples of abstractive generation in Table 1.

Specific details of unsupervised text generation models, which do not use labels for query-document pairs, are described in § 4.3.

Stochastic Generation While a naïve abstractive generation scheme can generate novel terms that are not included in the original document, a major drawback of this scheme is that they cannot generate a high volume of different terms for the given document. In other words, this scheme is suboptimal since it only generates a single sequence, though the terms within the document can have many synonymous expressions. To overcome this limitation, we stochastically generate terms for the given document by perturbing its embeddings for text generation via applying Monte Carlo (MC) dropout (Gal and Ghahramani, 2016). Compared to the abstractive generation scheme in Equation 5, which only produces one typical sequence of terms $\boldsymbol{t}_{abs}$, we obtain S different sequences $\boldsymbol{T}_{abs}$ from the stochastic generation scheme, as follows:

$$\begin{aligned} \boldsymbol{T}_{abs} &= \left[\boldsymbol{t}^i_{abs}\right]_{i=1}^{S} \\ \boldsymbol{t}^i_{abs} &= g'_{abs}(\boldsymbol{d}; \boldsymbol{\theta}_{abs}), \end{aligned} \tag{6}$$

where g'_{abs} randomly masks weights on the model even at test time. We provide examples of stochastic generation with $S = 3$ in Table 1. As shown in Table 1, examples of stochastic generation are more relevant to the document and more diverse.

4 Experimental Setups

Here, we describe datasets, models, evaluation metrics, and implementation details for experiments.

4.1 Datasts

We use two benchmark datasets for IR to evaluate our UDEG framework as follows:

ANTIQUE: This is a dataset with 403,666 documents from Yahoo! Answer, including open-domain non-factoid questions (Hashemi et al., 2020). The test set consists of 200 queries and 6,589 query-document pairs.

MS MARCO: This is a collection of 8,841,823 passages from Bing search engine (Nguyen et al., 2016). Since the test set is not publicly available, we use the development set containing 6,980 queries and 59,273 query-document pairs. We randomly sample 1,000,000 passages, while using the same development set for queries and query-document pairs, due to the limitation of computational resources on expanding 8,841,823 passages.

4.2 Retrieval Models

In this subsection, we describe two retrieval models that are widely used for IR systems.

BM25: This is one of the standard *ad-hoc* retrieval models based on Term Frequency-Inverse Document Frequency (TF-IDF), which measures overlapping terms between query and document (Robertson et al., 1994).

QL: This is also one of the standard *ad-hoc* retrieval models. Specifically, QL returns a ranked list of documents sorted by the probability of $P(d|q)$, where q is a query and d is a document (Zhai and Lafferty, 2017).

4.3 Expansion Models

We compare our UDEG framework against the following baselines:

No Expansion (No Expan.): This is a naïve model of retrieving the original documents without query or document expansion.

RM3: This is a query expansion model that uses a pseudo-relevance feedback scheme (RM3) (Jaleel et al., 2004). Note that this can be simultaneously used with document expansion models.

MP-rank: This is an extractive document expansion model, which extracts keyphrases based on a multipartite graph, where the nodes are keyphrase candidates and an edge connects nodes having different topics (Boudin, 2018).

LexRank: This is an extractive document expansion model that extracts the key sentence with PageRank algorithm (Page et al., 1998), which constructs vertices as sentences and edges as TF-IDF weights (Erkan and Radev, 2004).

PEGASUS$_{ext}$: This is an extractive document expansion model (Zhang et al., 2020a), which extracts sentences using pre-trained knowledge for generating masked sentences on the CNN/DailyMail dataset (Nallapati et al., 2016).

LexRank + paraphrase (Lex. + Para.): This is an abstractive document expansion model, which first extracts key sentences with LexRank, and then paraphrases them with an unsupervised model (Liu et al., 2020) based on simulated annealing.

UDEG: Our framework of expanding documents with abstractly generated sentences from a pre-trained language model. Diverse sentences are generated with stochastic perturbation by MC dropout.

		No Expan.	MP-rank	LexRank	Lex.+Para.	PEGASUS$_{ext}$	UDEG (Ours)
MRR	BM25	0.595	0.584	0.571	0.561	0.585	**0.645**
	BM25+RM3	0.558	0.579	0.542	0.567	0.555	**0.616**
	QL	0.499	0.534	0.567	0.518	0.562	**0.650**
	QL+RM3	0.396	0.447	0.456	0.432	0.504	**0.583**
R@10	BM25	0.218	0.220	0.208	0.209	0.207	**0.237**
	BM25+RM3	0.217	0.221	0.208	0.204	0.213	**0.226**
	QL	0.189	0.199	0.203	0.196	0.205	**0.232**
	QL+RM3	0.159	0.179	0.182	0.162	0.191	**0.211**
P@3	BM25	0.378	0.381	0.346	0.351	0.356	**0.431**
	BM25+RM3	0.361	0.355	0.360	0.373	0.366	**0.433**
	QL	0.301	0.333	0.340	0.315	0.358	**0.418**
	QL+RM3	0.240	0.281	0.275	0.271	0.301	**0.386**
MAP	BM25	0.211	0.212	0.199	0.202	0.201	**0.238**
	BM25+RM3	0.212	0.213	0.203	0.203	0.207	**0.234**
	QL	0.172	0.191	0.192	0.181	0.199	**0.230**
	QL+RM3	0.150	0.168	0.170	0.158	0.180	**0.212**
NDCG@3	BM25	0.437	0.442	0.417	0.425	0.419	**0.478**
	BM25+RM3	0.424	0.434	0.423	0.433	0.426	**0.470**
	QL	0.356	0.389	0.400	0.375	0.413	**0.471**
	QL+RM3	0.277	0.324	0.319	0.306	0.350	**0.424**

Table 2: Retrieval results on the ANTIQUE dataset. We use five evaluation metrics: MRR, R@10, P@3, MAP, and NDCG@3. Also, the best performance is marked in **bold**.

4.4 Metrics

We evaluate the models with five metrics, ranging from precision- to recall-oriented, as follows:

Mean Reciprocal Rank (MRR): MRR measures the location of the first relevant document for the given query in a binary sense.

Recall (R@K): R@K measures the recall up to K recommended documents.

Precision (P@K): P@K measures the precision up to K recommended documents.

Mean Average Precision (MAP): Similar to P@K, MAP evaluates all related documents with an ordered list of them.

Normalized Discounted Cumulative Gain (NDCG@K): Compared to the MAP that uses binary relevance metrics, this further manipulates the recommended list by using the fact that some documents are more relevant than others.

4.5 Implementation Details

All of the retrieval models are implemented using Anserini open-source IR toolkit (Yang et al., 2018) with the default hyperparameter values. The PEGASUS-large model, already fine-tuned on the XSUM dataset (Narayan et al., 2018), is used as a pre-trained language model in UDEG for abstractive text generation. For the decoding algorithm, we use a beam search algorithm and set the beam size as 8. Also, we set the number S of stochastic generation for document expansion as 4.

		No Expan.	LexRank	UDEG (Ours)
MRR	BM25	0.427	0.441	**0.463**
	BM25+RM3	0.366	0.385	**0.415**
	QL	0.402	0.420	**0.454**
	QL+RM3	0.319	0.337	**0.382**
R@10	BM25	0.636	0.646	**0.679**
	BM25+RM3	0.600	0.617	**0.651**
	QL	0.611	0.633	**0.671**
	QL+RM3	0.552	0.579	**0.629**
P@1	BM25	0.311	0.324	**0.344**
	BM25+RM3	0.248	0.265	**0.291**
	QL	0.289	0.302	**0.334**
	QL+RM3	0.202	0.215	**0.255**
MAP	BM25	0.422	0.435	**0.457**
	BM25+RM3	0.361	0.380	**0.409**
	QL	0.398	0.414	**0.448**
	QL+RM3	0.315	0.333	**0.377**

Table 3: Retrieval results on MS MARCO dataset. We use following evaluation metrics: MRR, R@10, P@1 and MAP. The best performance is marked in **bold**.

5 Results and Discussion

In this section, we show the overall performance of our UDEG, and then analyze the results in detail.

5.1 Overall Results

Results on the ANTIQUE dataset and sampled MS MARCO dataset are shown in Table 2 and Table 3, respectively. Our UDEG framework significantly outperforms all baselines in all evaluation metrics. Interestingly, the retrieval performance of QL is

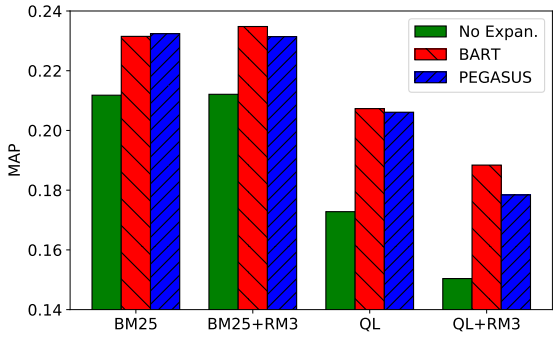

Figure 2: Comparison of BART and PEGASUS language models. The numbers of generated sentences for expansion are both set to one.

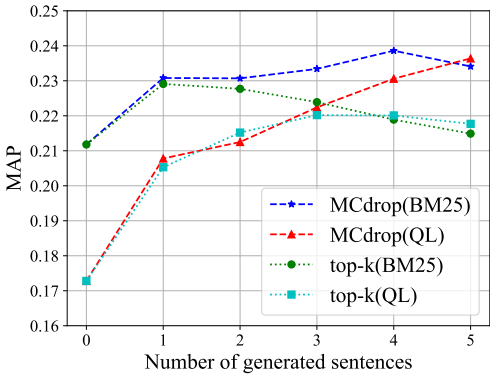

Figure 3: MAP scores of two different stochastic generation strategies (MC dropout vs. top-k sampling) with a varying number of generated sentences. When the number of generated sentences is 0, it refers to the naïve model without expansion.

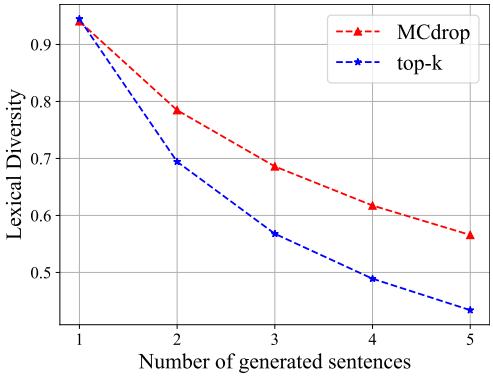

Figure 4: Lexical diversity of two different stochastic generation strategies (MC dropout vs. top-k sampling) with a varying number of expanded sentences.

impressively enhanced when using our framework. Note that the retrieval performance of QL without expansion is much lower than BM25; however, QL shows comparable and even outstanding performance with our expansion framework.

Effectiveness of Abstractive Generation Compared to the extractive and the paraphrasing baselines, our proposed abstractive framework outperforms them in all metrics. Notably, even though PEGASUS$_{ext}$ is pre-trained on the same PEGASUS pipeline with the UDEG framework, the expansion model with the extractive generation scheme is ineffective, since it cannot solve the vocabulary mismatch problem. However, the proposed UDEG framework can solve it by generating novel words, which demonstrates the effectiveness of the abstractive generation scheme.

Effectiveness of Query Expansion When RM3 is applied, the performance is negatively affected in most cases. As Nogueira et al. (2019) reported, we can also interpret the obtained results as evidence that document expansion is more effective than query expansion since a document often contains more signals than a query with its longer length.

5.2 Ablation and Discussion

Which attributes contribute how much to the performance improvement? To see this, we further perform an ablation study, as follows.

Robustness on Different Language Models To validate the robustness of our framework on different language models, we compare the performances of PEGASUS and BART (Lewis et al., 2020), both of which are trained on the XSUM dataset. As shown in Figure 2, the UDEG framework with PEGASUS shows performance similar to the one with BART, both of which consistently outperform the naïve baseline, which neither expands the query nor the document. Thus, the results show that the UDEG framework does not depend on a specific language model, but robustly improves the overall retrieval performance.

Comparison of Stochastic Generation Strategy We compare two stochastic generation strategies, MC dropout and top-k sampling. The top-k sampling is designed to generate diverse outputs by sampling the next word from the k most likely candidates, instead of deterministically selecting the next word (Fan et al., 2018). As shown in Figure 3, even though both strategies aim at generating diverse sentences stochastically, the MC dropout strategy outperforms the top-k sampling strategy. Where does this performance difference come from? The hypothesis is that MC dropout makes more diverse terms across sentences than top-k sampling. Specifically, we often obtain the same starting words from top-k sampling, which leads to generate a number of sentences that might share same starting words. On the other hand, MC

Query	How is the chemistry is a basic of science?
Relevant Document	Chemistry is a basic because all matter can be broken down into elements (i.e., hydrogen, oxygen, nitrogen, etc.); without matter, nothing could be studied.
Generated Sentences	**1)** Chemistry is the study of atoms and molecules. **2)** Chemistry is the study of matter and how it is made. **3)** Chemistry is the study of matter. **4)** Chemistry is a basic **science**.
	Original Document Rank: 104 Expanded Document Rank: 5
Query	How is the library consider as a heart of university?
Relevant Document	Whatever you are studying has to be found somewhere for you to learn it. That's where the library comes into focus.
Generated Sentences	**1)** If you're studying at **university**, you'll need a library. **2)** A library is a place where you can find out more about the subject you are studying. **3)** If you're studying, you'll be studying. **4)** There are many different ways you can study.
	Original Document Rank: 636 Expanded Document Rank: 32
Query	What do doctors do when a patient has a Do Not Resuscitate Order?
Relevant Document	All healthcare professionals involved in the care of that patient will not do anything to prolong the patient's life if in case patient deteriorates/dies. DNR orders may be modified, some may choose mechanical ventilation, or drugs. Usually when a pt is DNR, comfort measures is provided only.
Generated Sentences	**1)** DNR is not life-support. **2)** When a patient is in a "do not resuscitated" (DNR) state, that patient's life will not be saved. **3)** A DNR is a decision made by the patient's family or health care provider to prolong the life of the patient. **4)** A "do not resuscitate"(DNR) order does not mean that a patient should be put on life support.
	Original Document Rank: 40 Expanded Document Rank: 1

Table 4: Examples of generated sentences by the UDEG framework on the ANTIQUE dataset. Note that the first example contains scientific information. The generated terms are highlighted in red if the terms are novel but relevant to the document, and further highlighted in **bold** if the novel terms appear in the query.

dropout randomly perturbs the embeddings at the beginning of generating each sentence, which leads to a diversity of terms even at the starting point. To verify this hypothesis, we compare the lexical diversity of MC dropout and top-k sampling strategies with a varying number of generated sentences. The lexical diversity is calculated by averaging the proportion of the unique unigrams in generated sentences for each document. As Figure 4 shows, the lexical diversities of the generated sentences by top-k sampling are consistently lower and drop more rapidly than that by MC-dropout.

Varying the Number of Expanded Sentences To understand how stochastically generated sentences with MC dropout improves the retrieval performance, we experiment our UDEG with a varying number of generated sentences on two retrieval models, BM25 and QL. Figure 3 shows that the performances of both models tend to improve with increasing numbers of expanded sentences. Interestingly, QL is largely improved as stochastically generated sentences are stacked up to the original document. Meanwhile, the performance is slightly dropped when expanding five sentences for BM25. These results indicate that setting an appropriate number of generated sentences is important for optimal results, since too much information may degrade the context of the original document.

5.3 Case Study

For a qualitative analysis, we conduct a case study to explore the strengths of the UDEG framework. Table 4 shows examples of successfully retrieved expanded-documents with the UDEG framework compared to the original documents without expansion. Note that the original documents are retrieved with lower ranks, but get higher ranks after applying the UDEG framework. We note that the generated sentences contain novel words, while they sometimes contain copied terms. This tendency of copying increases the importance of the keyphrases which contributes to the effective term re-weighting. At the same time, newly generated terms are found to resolve the vocabulary mismatch problem by introducing synonyms or semantically related terms. These findings advocate for the importance of using abstractly generated sentences for document expansion in *ad-hoc* retrieval systems, which can help term re-weighting and alleviate the vocabulary mismatch problem at the same time.

6 Conclusion

We presented a novel framework, which we refer to as **U**nsupervised **D**ocument **E**xpansion with **G**eneration (UDEG), that generates diverse terms with stochastic perturbation over pre-trained language models, and efficiently enriches the document representation, without using any query infor-

mation for training. Remarkably, UDEG employed in a retrieval system shows significant performance improvements on two standard benchmark datasets. Also, a detailed analysis shows that an abstractive generation framework with stochastic perturbation positively contributes to the retrieval performance. Not only synonymy, but also other problems of the IR system such as polysemy could be addressed using our UDEG framework, to be left for the future work. We believe that the benefits of using diversely generated document-relevant sentences would allow further improvements on any IR system, targeting at scholarly and scientific information.

Acknowledgements

This work was supported by Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government MSIT) (No. 2018-0-00582, Prediction and augmentation of the credibility distribution via linguistic analysis and automated evidence document collection).

References

Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite graphs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 667–672. Association for Computational Linguistics.

Florian Boudin, Ygor Gallina, and Akiko Aizawa. 2020. Keyphrase generation for scientific document retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1118–1126. Association for Computational Linguistics.

Vincent Claveau. 2020. Query expansion with artificially generated texts. *arXiv preprint arXiv:2012.08787*.

Debasmita Das, Yatin Katyal, Janu Verma, Shashank Dubey, AakashDeep Singh, Kushagra Agarwal, Sourojit Bhaduri, and RajeshKumar Ranjan. 2020. Information retrieval and extraction on COVID-19 clinical articles using graph community detection and Bio-BERT embeddings. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Victor Dibia. 2020. Neuralqa: A usable library for question answering (contextual query expansion + BERT) on large datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 15–22. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.*, 22:457–479.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org.

Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W. Bruce Croft. 2020. Antique: A non-factoid question answering benchmark. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II*, volume 12036 of *Lecture Notes in Computer Science*, pages 166–173. Springer.

Nasreen Abdul Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah S. Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. 2004. Umass at TREC 2004: Novelty and HARD. In *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004*, volume 500-261 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Ashkan Kazemi, Verónica Pérez-Rosas, and Rada Mihalcea. 2020. Biased TextRank: Unsupervised graph-based content extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1642–1652, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer.

2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Xiangci Li, Gully A. Burns, and Nanyun Peng. 2020. A paragraph-level multi-task learning model for scientific fact-verification. *arXiv preprint arXiv:2012.14500*.

Jimmy Lin. 2019. The neural hype and comparisons against weak baselines. *SIGIR Forum*, 52(2):40–51.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2020. Unsupervised paraphrasing by simulated annealing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 302–312. Association for Computational Linguistics.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. *arXiv preprint arXiv:2005.00181*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Bhaskar Mitra and Nick Craswell. 2018. An introduction to neural information retrieval. *Found. Trends Inf. Retr.*, 13(1):1–126.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Karthik Narasimhan, Adam Yala, and Regina Barzilay. 2016. Improving information extraction by acquiring external evidence with reinforcement learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2355–2365. The Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

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. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Nikola I. Nikolov and Richard H. R. Hahnloser. 2020a. Abstractive document summarization without parallel data. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 6638–6644. European Language Resources Association.

Nikola I. Nikolov and Richard H. R. Hahnloser. 2020b. Abstractive document summarization without parallel data. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 6638–6644. European Language Resources Association.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 708–718. Association for Computational Linguistics.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

Jiho Noh and Ramakanth Kavuluru. 2020. Literature retrieval for precision medicine with neural matching and faceted summarization. *Findings of the Association for Computational Linguistics: EMNLP 2020*.

L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia.

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. *arXiv preprint arXiv:2010.08191*.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Min Joon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur P. Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4430–4441. Association for Computational Linguistics.

Cheng Tang and Andrew Arnold. 2020. Neural document expansion for ad-hoc information retrieval. *arXiv preprint arXiv:2012.14005*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

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

Jiacheng Xu, Shrey Desai, and Greg Durrett. 2020. Understanding neural abstractive summarization models via uncertainty. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6275–6281. Association for Computational Linguistics.

Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible ranking baselines using lucene. *ACM J. Data Inf. Qual.*, 10(4):16:1–16:20.

Ziyi Yang, Chenguang Zhu, Robert Gmyr, Michael Zeng, Xuedong Huang, and Eric Darve. 2020. TED: A pretrained unsupervised summarization model with theme modeling and denoising. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 1865–1874. Association for Computational Linguistics.

Chengxiang Zhai and John D. Lafferty. 2017. A study of smoothing methods for language models applied to ad hoc information retrieval. *SIGIR Forum*, 51(2):268–276.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Yong Zhang, Fen Chen, Wufeng Zhang, Haoyang Zuo, and Fangyuan Yu. 2020b. Keywords extraction based on word2vec and textrank. In *ICBDE '20: The 3rd International Conference on Big Data and Education, London, UK, April 1-3, 2020*, pages 37–42. ACM.