

RESEARCH

Open Access



Data augmentation for dense passage retrieval using corpus-passage frequency-based token deletion

A-Seong Moon¹, Kyumin Kim¹ and Jaesung Lee^{1,2*}

*Correspondence:
curseor@cau.ac.kr

¹ Department of Artificial Intelligence, Chung-Ang University, 84, Heukseok-ro, Dongjak-gu, Seoul 06911, Republic of Korea

² AI/ML Innovation Research Center, Chung-Ang University, 84, Heukseok-ro, Dongjak-gu, Seoul 06911, Republic of Korea

Abstract

This paper proposes a novel data augmentation method to address class imbalance in large-scale information retrieval systems. In particular, a corpus-passage frequency-based token deletion technique is introduced to improve the accuracy of Dense Passage Retrieval, which is a dense vector-based information retrieval model. Unlike traditional random token deletion methods that delete tokens with equal probability, the proposed method calculates token importance by considering both passage and corpus-level frequencies, leading to more effective token deletion. Experimental results demonstrate that the proposed approach significantly improves Top-k accuracy on smaller datasets compared to conventional augmentation techniques. While maintaining competitive performance on larger-scale datasets, its relative effectiveness is particularly notable in scenarios characterized by limited training data and severe class imbalance. This confirms its potential to improve the generalizability of information retrieval models. The source code is publicly available at https://github.com/asmoon002/DPR_TD.

Keywords: Information retrieval, Data augmentation, Natural language processing, Class imbalance

Introduction

Information retrieval systems have undergone significant advancements over the past few decades. Starting with basic keyword-based search methods, these systems have evolved into vector space-based models, probabilistic models, and, more recently, dense vector-based search techniques that incorporate deep learning and natural language processing. Despite these advancements, traditional sparse vector-based approaches continue to struggle with handling synonyms and semantically similar expressions, underscoring the need for more advanced retrieval algorithms.

Vector representation-based information retrieval systems can be classified into two categories: dense vector-based and sparse vector-based approaches. Traditional methods, such as TF-IDF and BM25 [1], construct vectors based on all unique words within a document collection, resulting in particularly high-dimensional vectors corresponding to large corpora. Because most documents utilize only a small subset of the vocabulary,

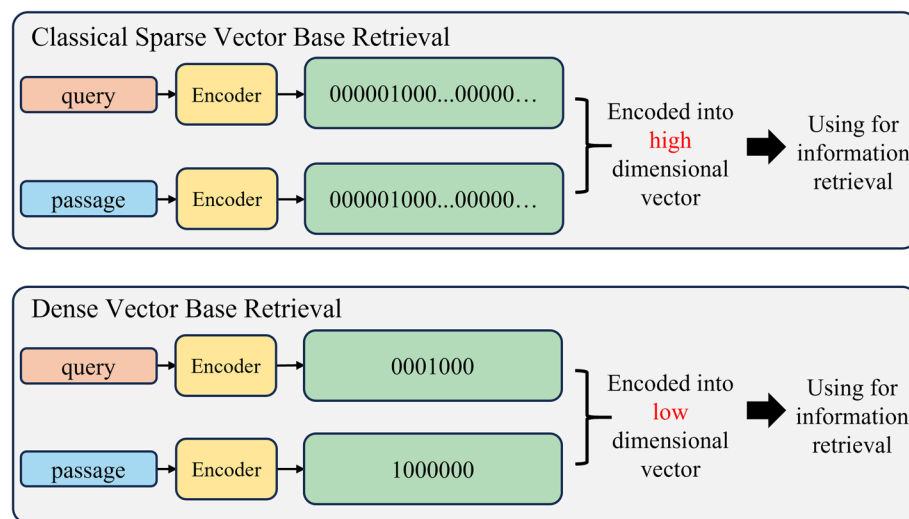


Fig. 1 Sparse vector-based retrieval and dense vector-based retrieval mechanisms

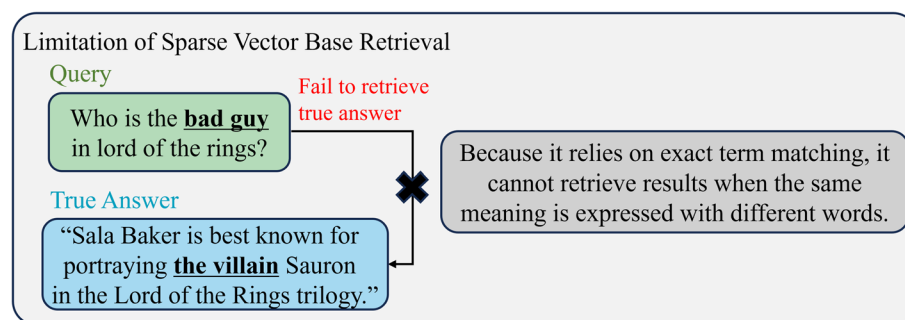


Fig. 2 Vocabulary mismatch problem in sparse vector-based retrieval. Sparse vector-based retrieval systems, which rely on exact term match, often fail the retrieval process. This example is adapted from [2]

the majority of vector dimensions remain sparse. Figure 1 illustrates the difference in density over different vector dimensions. Sparse vector-based retrieval systems rely on exact word matches or mismatches to identify similar documents. These systems often fail to recognize documents containing synonyms or related terms, considering them to be irrelevant. For example, if a query includes the term “bad guy” and a document uses “villain” in the same context, the sparse vector-based system may fail to retrieve the relevant document due to the lack of an exact term match, as depicted in Fig. 2 [2].

To overcome this limitation and ensure greater flexibility in information retrieval, dense vector representation has been introduced as an alternative approach. Dense passage retrieval (DPR) [2] employs large language models as encoders to transform both questions and documents into dense vector representations. These encoders are trained to maximize the similarity scores between question-answer pairs. Using dense vector representations, retrieval systems can address the limitation of exact word matching, thereby enabling the inclusion of synonyms and related terms. This significantly improves the likelihood of retrieving accurate results. In datasets used for question-answering tasks, most documents do not contain the correct answer to any given

question, and this imbalance becomes more pronounced as the dataset size increases. ‘Positive’ documents, i.e., those containing the correct answer, constitute only a small fraction compared to the large volume of unrelated ‘negative’ documents. This imbalance frequently causes models to focus heavily on patterns from negative examples, with limited opportunities to learn from positive examples, thereby reducing the overall learning performance of the model [3].

Data augmentation techniques are often employed to address the challenges induced by class imbalance. Existing methods, such as random token deletion, are valued for their simplicity and efficiency. However, these techniques often fail to preserve important information and reduce the quality of augmented data. To address this shortcoming, a corpus-passage frequency-based token deletion method is introduced in this study. The proposed approach aims to enhance retrieval accuracy by removing tokens selectively based on their significance. This method retains meaningful content while discarding unnecessary data by distinguishing between essential and less critical information. It addresses the data imbalance using data augmentation, thereby enhancing the ability of the model to learn from positive examples and improve retrieval accuracy. Unlike traditional random token deletion techniques, this approach enhances the overall performance of information retrieval models.

The main contributions of this paper are as follows:

- A novel token deletion technique is proposed that leverages both corpus and passage frequency to remove less important tokens selectively while preserving critical information for DPR.
- The proposed method addresses the issue of class imbalance by applying targeted data augmentation, enhancing the ability of models to learn from positive examples, and improving their retrieval accuracy.
- Experimental results demonstrate that the frequency-based token deletion approach outperforms traditional random deletion methods in terms of retrieval efficiency and accuracy.

Related works

Dense passage retrieval

DPR was developed to address the limitations inherent in traditional sparse vector-based information retrieval methods such as TF-IDF and BM25, which depend on exact keyword matches [4]. Sparse methods struggle to identify synonyms or semantically similar expressions [5, 6]. To overcome these issues, DPR encodes both questions and documents into dense vectors that capture semantic relationships effectively [7]. At the core of DPR lies a dual-encoder architecture consisting of two separate encoders that transform questions and documents into dense vector representations [8]. A pre-trained language model based on Bidirectional Encoder Representations from Transformers (BERT) [9] is commonly utilized for these encoders. BERT serves as the central encoder, converting input sequences into dense representations by leveraging bidirectional context. This enables it to attend to words on both sides of the target word simultaneously, thereby yielding comprehensive contextual embeddings [9]. During the encoding

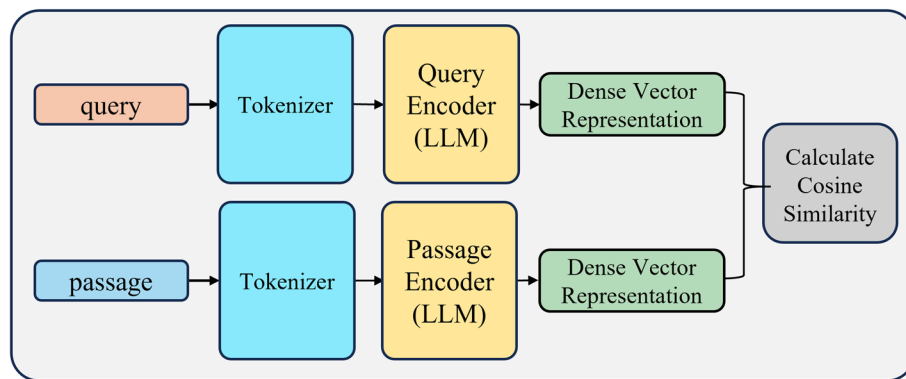


Fig. 3 Overview of the dense passage retrieval process

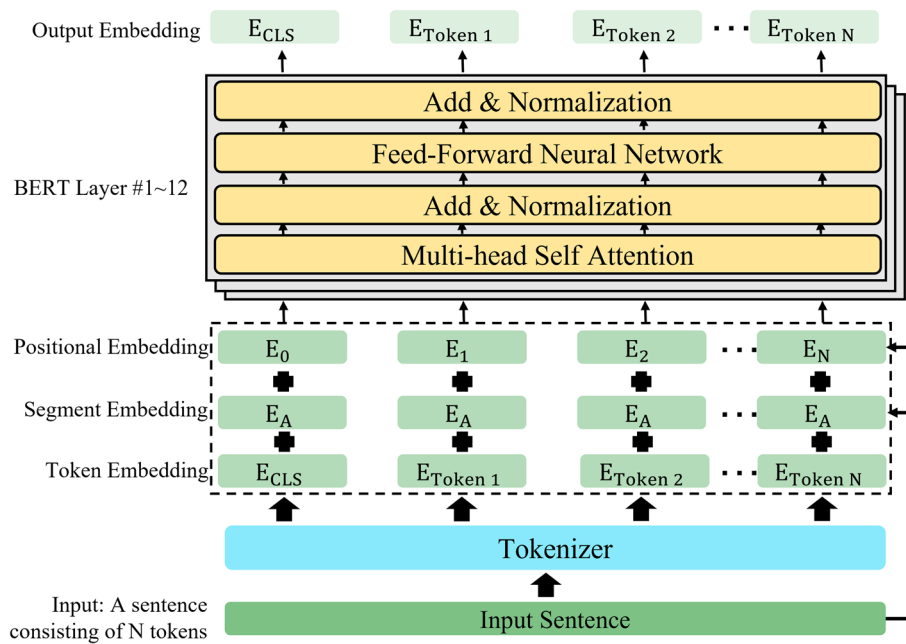


Fig. 4 Overview of BERT architecture. BERT is utilized as the encoder in the DPR structure of this paper. Input sentences are processed using a tokenizer to obtain token embeddings. Using the {CLS} token. Vector representations of entire sentences are derived using {CLS} tokens. Segmentation embedding is used to distinguish different sentences. This figure illustrates an example involving only one sentence. Positional embedding represents the order of tokens within a sentence. These three types of embeddings are combined for each token and then transmitted as input into the neural network layers. Each layer consists of Multi-head Self-Attention, Add & Normalization, and a Feed-Forward Neural Network (FFNN). The layers, referred to as BERT layers, are stacked into a total of 12 layers. After passing through these 12 layers, output embeddings for each token are obtained, and the embedding of the {CLS} token is considered to be the dense vector representation of the sentence

process, BERT uses subword tokenization techniques, such as WordPiece [10], and prepends a {CLS} token to the sequence, which serves as a special aggregation token summarizing the entire input.

Figure 3 illustrates the DPR process, and Fig. 4 provides a detailed representation of an input sequence being processed through the multiple layers of BERT [9]. Each layer

employs self-attention mechanisms that allow words to interact bidirectionally, capturing relationships between various tokens throughout the passage. The lower layers of BERT primarily capture syntactic relationships, whereas the upper ones capture higher-level semantic information [11]. The self-attention mechanism [12] generates Query (Q), Key (K), and Value (V) vectors for each token, which are used to calculate attention scores. These scores determine the level of focus placed on other tokens while processing a specific token. Outputs from multiple attention heads are concatenated and transformed to produce the final attention output of each layer. This approach enables the model to aggregate information corresponding to different positions in the sequence, thereby facilitating a comprehensive understanding of bidirectional contextual relationships [11].

Figure 4 shows the architecture of a BERT layer, highlighting its primary components: the multi-head self-attention mechanism, the position-wise feedforward neural network (FFNN), residual connections, and layer normalization [9]. The multi-head self-attention mechanism allows the model to focus on various aspects of the input simultaneously by employing multiple attention heads, with each capturing different relationships between the words [12]. Subsequently, the FFNN introduces non-linearity via two linear transformations separated by an activation function, further refining the information [13]. Residual connections are added to preserve the original input across layers, thereby enhancing training stability. Layer normalization ensures a stable input distribution. As the input progresses through the BERT layers, the {CLS} token accumulates contextual information from the entire sequence. At the top-most layer (Layer #12 in the diagram), the final representation of the {CLS} token is generated, creating a dense vector that captures the semantic content of the input. This vector serves as a concise summary of the meaning of the entire passage or question. Together, these components enable the model to learn intricate patterns and capture contextual relationships at multiple levels of its stacked-layers structure [14]. This architecture equips DPR with the ability to generate meaningful, dense vector representations, which are crucial for natural language processing tasks, and enables it to outperform traditional sparse retrieval methods [15]. In DPR, dense DPR-encoded vectors are utilized to assess the similarity between questions and documents. DPR calculates similarity scores using certain techniques, e.g., taking the dot product after representing both the question and the passage as dense vectors [16]. Based on training on large-scale datasets, BERT encoders refine their ability to yield effective vector representations, thereby improving retrieval performance by accurately capturing the semantic relationships between queries and relevant documents.

Figure 5 describes the entire dual encoder architecture used in DPR. In this framework, one encoder functions as the question encoder, denoted by $E_{Q(q)}$, while the other serves as the passage encoder, denoted as $E_{P(p)}$. The question and passage encoders transform the input question and document, respectively, into dense 768-dimensional vectors. The similarity between these vectors is calculated using the dot product, which facilitates the retrieval of the most relevant documents.

$$\text{sim}(q, p) = E_{Q(q)}^\top E_{P(p)}$$

Using dot product to calculate similarity scores enables efficient retrieval, particularly on large-scale datasets, by leveraging maximum inner product search (MIPS) algorithms [17].

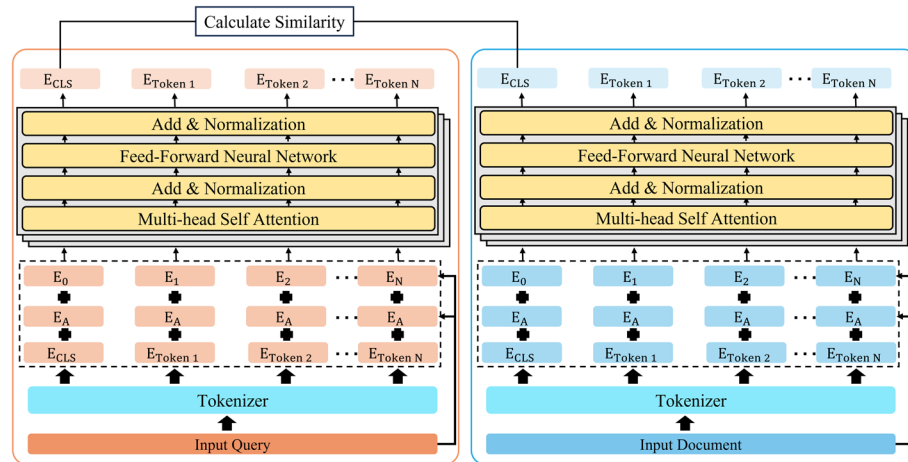


Fig. 5 Overview of dual-encoder architecture, including two neural network encoders. In this paper, BERT is employed. When a query and a document are provided as input, one encoder generates an embedding for the query, while the other generates an embedding for the document. During training, the parameters of the two encoders are fine-tuned independently. The similarity between the two inputs is measured by calculating the similarity between the {CLS} token outputs obtained from each encoder

By design, the dual-encoder architecture exhibits efficient computation by separating the processing of questions and passages. Each encoder operates independently and generates dense vectors that map both queries and passages to a shared semantic space. This design enhances scalability because dense passage representations can be precomputed and stored, enabling rapid retrieval using approximate nearest neighbor search [18] techniques, such as MIPS. Furthermore, independent encoders can be fine-tuned using distinct objectives, allowing for flexible optimization across various tasks.

In DPR, several strategies are available for selecting negative passages to address class imbalance [19]. One approach involves random negative selection, where irrelevant passages are chosen randomly from the entire corpus. However, these often lack sufficient similarity to the query, making them less challenging for the model and resulting in inefficient learning [20]. An alternative and more effective strategy involves the use of BM25-based negatives. In this method, the top-ranking passages retrieved by the BM25 algorithm, i.e., those matching most of the query tokens but excluding the correct answer, are selected. These negatives, known as “hard negatives”, exhibit greater contextual similarity to the correct answer, compelling the model to refine its ability to distinguish between relevant and irrelevant passages. However, the BM25-based negative selection method is computationally intensive and difficult to scale when applied to large datasets, such as Wikipedia or Common Crawl. To enhance training efficiency, in-batch negative sampling is utilized frequently in DPR. In this method, negative examples corresponding to a given question are selected from other questions within the same mini-batch, leveraging the diversity of batches effectively to generate a broad set of negative examples without relying on additional negative sampling methods. Although this approach significantly reduces computational demands, it requires careful parameter tuning to prevent overwhelming the model with an abundance of negative examples [21].

In conclusion, although DPR exhibits significant advantages over traditional sparse retrieval methods, class imbalance remains a major obstacle. Effective data augmentation strategies are crucial for minimizing the risk of overfitting to negative examples. However, these solutions frequently involve trade-offs in terms of scalability, computational cost, and overall complexity.

Conventional data augmentation

Data augmentation is a technique commonly used to expand the size and variety of training datasets used in machine learning by generating modified versions of existing data. In fields such as image and audio processing, which involve continuous data, methods such as cropping, rotation, and noise addition are frequently employed [22]. These transformations enhance the generalizability of models by exposing them to different input data variations. However, augmenting text data poses unique challenges due to its discrete nature. Unlike images, in which small modifications typically preserve the overall structure, even minor text changes can significantly alter its meaning or context [23]. Consequently, conventional augmentation techniques used for image or audio processing are not directly applicable to text data [24].

To address these limitations, several augmentation methods have been developed specifically for text. One notable approach is Easy Data Augmentation (EDA) [25], which has led to the design of four straightforward yet effective techniques for text data. The first technique, Synonym Replacement (SR), replaces selected words in a sentence randomly with their synonyms. This approach enhances model generalizability by incorporating lexical diversity while preserving the core meanings of sentences. The second technique, Random Insertion (RI), introduces synonyms into random positions within a sentence, which adds noise and motivates the model to handle more diverse sentence structures. The third technique, Random Swap (RS), randomly switches the positions of pairs of words in each sentence to create variations in word order. Finally, Random Deletion (RD) removes words from the sentence with a specified probability, encouraging the model to infer meaning based on incomplete information, thereby improving its generalizability [25]. Table 1 provides examples of these EDA techniques.

The EDA approach has been demonstrated to enhance performance effectively in various text- classification tasks, particularly on small datasets. A previous study [25] indicates that EDA-based methods increase accuracy by 3% on average on datasets containing as few as 500 training samples. Remarkably, even when using only 50%

Table 1 Illustration of easy data augmentation (EDA)

Operation	Sentence
None	A sad, superior human comedy played out on the back roads of life.
Synonym replacement	A <i>lamentable</i> , superior human comedy played out on the <i>backward</i> roads of life.
Random insertion	A sad, superior human comedy played out on <i>funniness</i> the back roads of life.
Random swap	A sad, superior human comedy played out <i>roads</i> back <i>the</i> of life.
Random deletion	A sad, superior human on the back roads of life.

of the data, EDA improves accuracy by an extent comparable to that of training on the complete dataset. This makes it especially valuable in low-resource environments, where obtaining large quantities of labeled data is difficult.

Among these techniques, RD has garnered significant attention because of its simplicity and effectiveness. By randomly removing words from sentences, it compels the model to interpret incomplete data, thereby improving its robustness [26]. Ablation tests within the EDA framework have demonstrated that RD often outperforms other augmentation methods, such as SR or RI, particularly on very small datasets [25]. This advantage can be attributed to the greater benefit enjoyed by RD in low-data scenarios, stemming from its generalizability gained by recognizing patterns in incomplete text rather than relying on slightly altered versions of the original data. These findings suggest that RD serves a dual purpose: a data augmentation strategy and a form of regularization by introducing controlled noise during training to reduce overfitting. This approach is particularly advantageous in low-resource environments with limited labeled data, enabling models to be effectively generalized using fewer training samples. Recent studies [25, 27] have further highlighted that RD consistently achieves better performance on small datasets compared to other techniques by promoting the learning of generalized patterns instead of reusing specific instances.

RD enhances model generalizability based on remaining data, thereby improving its robustness to input variations. When implemented at the token level instead of the word level, RD operates with greater precision, offering particular advantages for transformer-based models, such as BERT [9] and generative pre-trained transformers [28], which utilize subword tokenization. These models are designed to capture contextual relationships across tokens. The deletion process enables the adjustment of the attention mechanism, further enhancing the ability of the model to differentiate between important and less relevant tokens. Figure 6 depicts an example of tokenized sentences demonstrating this approach.

However, despite its advantages, random token deletion faces several challenges. One of the most critical issues is the indiscriminate nature of the deletion process. Although removing tokens can enhance model generalizability, it also carries the risk of eliminating crucial tokens. This can distort the meaning of a sentence and lead to incorrectly labeled augmented data. For instance, in the sentence, “The medicine is not harmful”, removing the word “not” entirely reverses its meaning, potentially causing misclassification. Such errors can lead the model to learn incorrect patterns, which may adversely affect its overall performance [29].

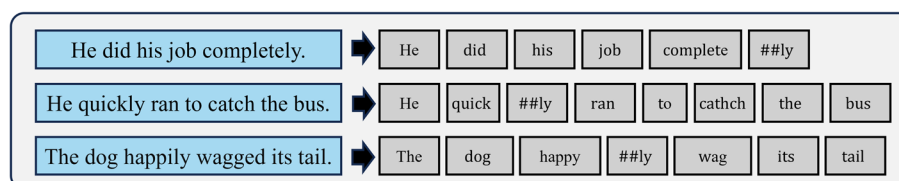


Fig. 6 Example of tokenized sentences. The tokenizer generates tokens for the input sentence, and stemming is performed on each token to separate prefixes and suffixes further. Prefixes or suffixes are marked by adding “##” at their respective positions

Another significant challenge is the potential loss of context, particularly in complex or domain-specific texts [30–33]. On specialized datasets where precise terminology and correct word order are critical, random token deletion can inadvertently remove key information essential for understanding a sentence. This issue is particularly problematic in certain tasks, such as medical text analysis or legal document classification, where even the deletion of a single token can drastically alter the outcome. Furthermore, RD can introduce noise into the training data when a significant number of important tokens are removed, potentially confusing the model rather than enhancing its generalizability.

Further, random token deletion may be counterproductive for shorter sequences, as deleting multiple tokens from such sequences can result in overly sparse sentences that do not convey meaningful information. Whereas longer sentences can withstand the removal of a few tokens without losing coherence, shorter sentences are far more sensitive because even the deletion of one or two tokens can significantly alter their meaning. This issue is compounded in imbalanced datasets, where RD of tokens from minority class examples may inadvertently remove key features, further hindering the ability of the model to learn accurate representations of these classes.

To address these challenges, a more advanced method, i.e., selective token deletion, is introduced in this paper. In this approach, the significance of each token is evaluated before deciding whether to delete it, often by calculating its contribution to the overall meaning of the sentence. Although this technique can preserve the essential meaning of sentences, it introduces additional computational costs and complexity, thereby reducing the simplicity and efficiency that make of basic RD.

An Easier Data Augmentation (AEDA) [34] is a text data augmentation technique developed to enhance text classification performance by inserting punctuation marks into sentences. The set of punctuation marks used is {".", ";", "?", ":", "!", ","}. Unlike conventional data augmentation methods such as EDA, which rely on word-level operations, such as substitution, deletion, or insertion, AEDA employs a simpler and less intrusive approach.

The central idea of AEDA is to preserve the original content and word order while introducing small changes by adding punctuation marks. This method aims to maintain the meaning and structure of the sentence, thereby minimizing the risk of altering the text labels or introducing misleading information. By applying minimal modifications, AEDA improves model generalizability, particularly in situations where labeled data is scarce. Table 2 provides examples illustrating the application of AEDA.

Although AEDA offers a straightforward and less intrusive approach to data augmentation, it has some limitations. Its main drawback is that the insertion of punctuation marks restricts its ability to generate highly diverse training examples. In tasks where

Table 2 An example of an easier data augmentation (AEDA)

Case	Sentence
Original text	A sad, superior human comedy played out on the back roads of life.
Augmentation sample 1	A sad, superior. human ! comedy played ; out on the back , roads of life.
Augmentation sample 2	A sad, . superior ; human comedy ? played out on the : back roads of life.
Augmentation sample 3	: A sad, superior , human comedy . played out . on the back ? roads , of life.

improving model performance requires substantial variation in data, the relatively subtle modifications introduced by AEDA may struggle to provide sufficient diversity, potentially limiting its effectiveness in preventing overfitting. Further, as the number of augmentations is increases, the tendency of the augmented examples to exhibit similar patterns also increases, imposing constraints on the number of meaningful augmentations that can be generated.

Additionally, although AEDA may perform well in sentence-level data augmentation, its limitations become more noticeable as the number of sentences increases. Because AEDA functions by inserting punctuation marks randomly, the augmented data can occasionally appear to be unnatural or noisy in a way that fails to provide meaningful information during model learning. This issue can be particularly problematic in scenarios where punctuation is critical for conveying the meaning of a sentence, as the inserted punctuation marks could alter the intended interpretation and confuse the model.

Table 3 provides examples in which AEDA can affect the text negatively. Because the original text naturally includes some punctuation marks, excessive or repetitive insertion of punctuation can disrupt the contextual meaning between sentences in this case. For instance, in the example “When punctuation marks are placed after every word”, all six types of punctuation marks are inserted after each word, resulting in unnatural and fragmented text. Similarly, in the example “When a period is placed after every word”, each word becomes isolated as a separate sentence, transforming the original content into a simple list of words. This outcome highlights the failure to retain the meaning of the original text and illustrates a potential pitfall of AEDA.

Recent advancements in data augmentation for dense passage retrieval have introduced sophisticated techniques that utilize external resources such as pre-trained language models (LLMs), synonym dictionaries, and structured knowledge bases. RocketQA [35] improves retrieval accuracy through a dual-encoder framework supported by

Table 3 Bad cases of an easier data augmentation(AEDA)

Case	Sentences
Original text	Waiting for a Girl Like You “Waiting for a Girl Like You” is a 1981 power ballad by the British-American rock band Foreigner. The distinctive synthesizer theme was performed by the then-little-known Thomas Dolby, and this song also marked a major departure from their earlier singles because their previous singles were mid to upper tempo rock songs while this song was a softer love song with the energy of a power ballad. It was the second single released from the album “4” (1981) and was co-written by Lou Gramm and Mick Jones.
When punctuation marks placed after every word	Waiting, for; a, Girl! Like, You? Waiting; for: a, Girl, Like; You! is, a, 1981; power, ballad! by, the: British-American; rock, band! Foreigner. The, distinctive; synthesizer: theme? was! performed, by, the; then-little-known: Thomas, Dolby! and, this: song? also; marked, a! major, departure: from; their, earlier! singles? because; their, previous: singles! were, mid, to: upper; tempo? rock, songs: while! this, song; was! a: softer; love? song, with: the! energy, of; a, power? ballad! It, was; the: second, single; released? from: the! album, 4; 1981! and? was, co-written: by, Lou; Gramm! and, Mick: Jones;
When a period is placed after every word	Waiting. for. a. Girl. Like. You. “Waiting. for. a. Girl. Like. You.”. is. a. 1981. power. ballad. by. the. British-American. rock. band. Foreigner.. The. distinctive. synthesizer. theme. was. performed. by. the. then-little-known. Thomas. Dolby,. and. this. song. also. marked. a. major. departure. from. their. earlier. singles. because. their. previous. singles. were. mid. to. upper. tempo. rock. songs. while. this. song. was. a. softer. love. song. with. the. energy. of. a. power. ballad.. It. was. the. second. single. released. from. the. album. “4”. (1981). and. was. co-written. by. Lou. Gramm. and. Mick. Jones..

The repetitive insertion of punctuation marks in AEDA disrupts the original sentence structure. Furthermore, in passages consisting of multiple sentences, it breaks the logical connections between sentences, thereby harming the overall context

cross-encoder-based distillation. MoMA [36] introduces memory-oriented augmentation by leveraging external memory modules to enhance retrieval under low-resource settings. More recently, ConvAug [37] proposes a conversational augmentation strategy that utilizes LLMs to synthesize user-like queries and improve robustness in dialogue-based retrieval. SyNeg [38] leverages LLMs to synthesize contextually rich hard negatives for dense retrieval. These recent works represent a growing trend toward external resource-intensive augmentation pipelines that offer strong retrieval performance but require significant computational overhead and complex infrastructure. They are particularly effective in large-scale or high-resource environments but may not generalize well in constrained domains where access to LLMs, external data, or robust training infrastructure is limited.

In contrast, our method seeks to provide a lightweight, interpretable, and corpus-internal augmentation strategy by identifying statistically redundant tokens based on Passage Frequency and Corpus Frequency. This focus enables us to enhance passage discriminability without relying on external models or linguistic resources. The design aligns with the principles of simplicity and self-containment, which have underpinned classical methods like EDA and AEDA, while addressing their limitations in scalability and contextual awareness. By focusing on internal statistics and avoiding reliance on LLMs or external data, the proposed approach complements existing trends in augmentation research, offering a practical alternative for settings where computational efficiency, interpretability and domain isolation are key concerns.

Proposed method

Problem definition

Data augmentation has emerged as a crucial tool in machine learning, particularly in text-based tasks, where it is used to expand training datasets to enhance model generalizability and address challenges such as overfitting and class imbalance [39]. Augmentation plays a crucial role in enhancing model learning in fields such as information retrieval, where tasks often involve large-scale corpora. Figure 7 depicts an example of

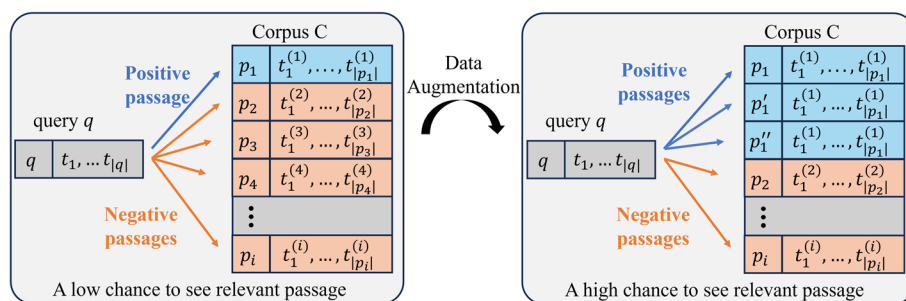


Fig. 7 Illustration of data augmentation used to address the shortage of positive passages. The left section illustrates a common problem situation often found in traditional QA datasets. Among passages, only one highly relevant positive passage is considered to provide the correct answer for a given query. In this scenario, the number of negative passages exceeds the number of positive passages, causing the model to encounter negative passages more frequently than positive passages during training. This issue is addressed by increasing the number of positive passages using data augmentation, as illustrated in the right section. Even though there may be passages that can be considered positive despite not having the exact answer, their proportion is still significantly smaller compared to negative passages

data imbalance. However, traditional methods, such as RD, and simple augmentation approaches, such as EDA and AEDA, suffer from inherent limitations that can diminish their effectiveness.

One major limitation of RD is the indiscriminate approach to text modification. Such methods delete tokens with equal probability, irrespective of their contextual significance or contribution to the overall meaning of a passage. This can result in the removal of critical tokens that are essential to conveying the meaning, thereby distorting sentences and reducing the quality of the augmented data. For example, in an information retrieval task, deleting a key token from a passage, such as “Lincoln” in the context of “Who was the 16th President of the United States?” could render the passage irrelevant and impair retrieval accuracy [40]. This issue is particularly pronounced in domains that require precise language formulations, such as legal and medical texts [30, 31]. Figure 8 illustrates the challenges associated with the traditional EDA token deletion method, in which all tokens are removed with equal probability.

RD introduces significant risks in imbalanced datasets, which represent a common challenge in information retrieval tasks where positive passages (containing the correct answer) are outnumbered by negative ones (not containing the correct answer). In such cases, the indiscriminate removal of tokens from already limited positive examples can exacerbate class imbalance. This can result in reduced learning from positive examples, further impairing the ability of the model to retrieve correct passages during inference [40]. RD also risks inadvertently removing tokens that are critical for establishing relevance, which becomes especially problematic in sparse or low-resource datasets.

To address these limitations, we introduce a corpus-passage frequency-based token deletion method in this study. This approach removes tokens selectively, considering their frequencies within individual passages and across the entire corpus. By calculating token importance using both passage and corpus-level frequencies, it prioritizes retaining meaningful tokens and keywords while reducing the emphasis on less informative tokens, such as common stop words. By prioritizing the preservation of important tokens, the proposed method aims to enhance retrieval performance by generating

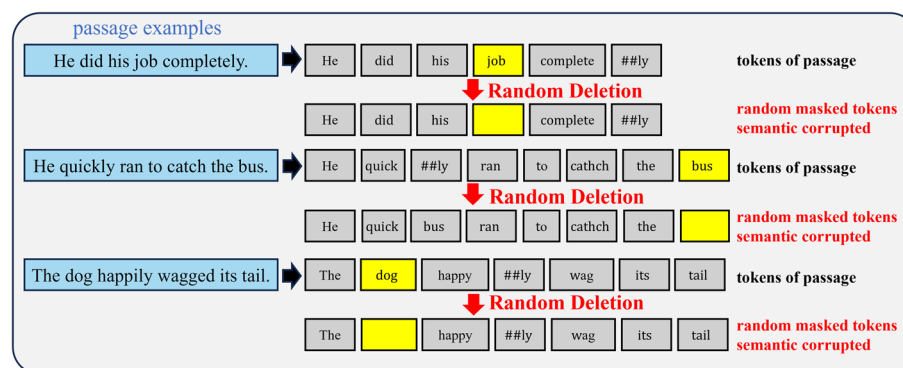


Fig. 8 Example of EDA base random deletion. The deletion method in traditional EDA removes tokens with the same probability for all tokens. As a result, tokens that play a crucial role in preserving the meaning of the sentence may be removed. For instance, in the sentence “He did his job completely” as depicted in the example, if the token “job” is deleted, the meaning is completely obscured, weakening the intended meaning of the sentence

higher-quality augmented data, thereby effectively mitigating the limitations of indiscriminate token deletion.

Weight-based token deletion method

By refining the data augmentation process as discussed in the preceding section, the proposed method overcomes the limitations of earlier approaches and offers a robust solution to address data scarcity. This approach enhances the ability of the model to learn from augmented data while ensuring that the essential elements of the text, which are crucial for accurate question answering, are preserved. As a result, the proposed method enhances the overall performance and reliability of large language models used in information retrieval and other natural language processing tasks.

Figure 9 presents an illustrative example of the proposed method, which performs token deletion based on the product of Passage Frequency (PF) and Corpus Frequency (CF). In this example, the token “##ly” appears in all three example passages, making it the token with the highest $PF \times CF$ score in the illustrated case. As a result, it is assigned the highest deletion probability. Each passage is tokenized, and tokens with the highest score are deleted to produce semantically preserved variations. For example, in the second passage, the token “quickly” is split into “quick” and “##ly”, and only “##ly” is removed. This reflects the model’s ability to reduce redundancy while retaining core semantics.

PF is mathematically represented by Equation (1), where D represents an individual document within the document set, and f denotes a frequency function that enumerates the occurrence of a term in a specific passage.

$$PF(t, p_i) = \frac{f_{t,p_i}}{\sum_{p \in D} f_{t,p_i}} \quad (1)$$

This is statistical measure indicates how frequently a token appears within a single passage.

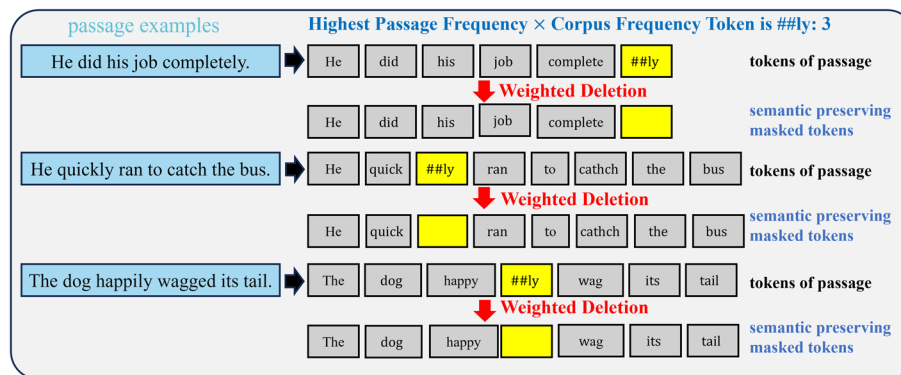


Fig. 9 Example of proposed weighted deletion. The proposed Weighted Deletion method determines the deletion probability of each token by calculating its importance based on phrase frequency and corpus frequency. In this figure, the corpus frequency and phrase frequency are computed in a scenario involving three sentences. The figure demonstrates that the “##ly” token exhibits the highest weight, resulting in the highest deletion probability, and demonstrates the token being removed

CF, defined by Equation (2), quantifies the frequency of occurrence of a token within the entire corpus of passages.

$$CF(t, C) = \log \frac{|p_i \in C : t \in p_i|}{|C|} \quad (2)$$

Although the definitions of PF and CF resemble the traditional term frequency (TF) and the inverse document frequency (IDF), their interpretations in this study differ due to the distinct objective of token-level data augmentation for DPR. In traditional IR, a high TF indicates that a term is central to the theme of a document, and a high IDF suggests a strong discriminative power between documents. However, in the context of DPR, where the goal is to distinguish passages that answer specific queries, such interpretations are not directly applicable.

It is important to clarify the rationale for using the geometric mean (GM) of PF and CF to determine token deletion probabilities. In our task-specific context, the objective is not merely to retain domain-relevant tokens but to enhance the discriminative capacity of passages during retrieval. For instance, consider the token “synth-pop” in a music corpus. If it exhibits both high PF and high CF, it frequently appears within many passages and is also repeated within individual passages. Despite being domain-specific, such a token fails to provide distinctive value for identifying relevant passages, as it represents the general background of the corpus rather than a unique signal. Deleting such tokens promotes a more focused retrieval process.

The GM formulation supports this strategy by assigning high weights (i.e., high deletion probabilities) only when both PF and CF are high—indicating that the token is redundant both locally and globally. Conversely, suppose either PF or CF is low. In that case, the GM value is suppressed, preserving tokens that are locally salient or globally rare, which are likely to be helpful in distinguishing passages. This design ensures that the augmentation process removes broadly redundant tokens while preserving contextually meaningful ones. It is thus consistent with the aim of optimizing dense passage retrieval rather than merely maintaining domain vocabulary.

$$Weight(t, p, C) = GM(PF(t, p), CF(t, C)) \quad (3)$$

Higher weights represent tokens that are less important to the specific context of each passage, thereby enabling more informed and targeted deletion during data augmentation. This strategy ensures that the essential tokens, which provide crucial information on correct answers, are less likely to be deleted, thereby preserving the integrity of the passage while effectively augmenting the data. By adjusting the deletion probability based on these weights, the proposed method enhances the training process of the model, resulting in improved and more reliable information retrieval performance.

In the information retrieval domain, tokens appearing frequently within individual passages and throughout the entire corpus are generally considered less informative. This is primarily because such tokens often consist of common words or phrases that lack significant semantic value in the context of distinguishing one passage from another. These tokens, commonly referred to as ‘stopwords’ include words such as “the”, “is”, “and” or “of” which appear in the majority of documents irrespective of the topic or content. Although these words are vital for maintaining the grammatical structure of a

sentence, they contribute little to identifying the unique characteristics or meaning of specific passages. For example, in a search query related to “climate change”, common words such as “the” or “and” are unlikely to help in identifying passages that are more relevant to the topic. On the other hand, terms that are more specific to the subject, such as “greenhouse gases” or “global warming” play a more important role in retrieving relevant documents. Consequently, frequently occurring tokens tend to introduce noise rather than provide meaningful information when queries are matched to relevant passages.

However, traditional stopwords removal is not directly applicable to dense retrieval models such as DPR. Unlike sparse retrieval methods, DPR relies on contextual embeddings produced by transformer-based encoders, where even seemingly non-informative tokens contribute to the overall semantic representation. Removing functional words such as prepositions or articles may disrupt the grammatical and semantic integrity of the passage, potentially harming retrieval performance. For example, eliminating the word “of” from the phrase “President of the United States” significantly alters its readability and meaning. Therefore, blanket removal of predefined stopwords may degrade the effectiveness of dense representations.

The proposed approach extends beyond simply identifying and removing traditional stopwords. It considers the overall characteristics of an entire corpus to evaluate the significance of individual tokens. Analyzing token frequency at both passage and corpus levels offers a more refined approach to assessing token importance, making it suitable for a wide range of domain-specific datasets. For example, let us consider a corpus that focuses on pop music. In this domain, tokens such as “pop”, “singer” or “song” are likely to appear frequently in multiple passages because they are integral to the overarching topic. Although these terms are central to the general subject matter, they do not contribute much to distinguishing passages within the same corpus. Their frequent appearance diminishes their relevance as indicators of unique content, making them less effective in identifying the most pertinent documents in response to specific queries. In contrast, terms that are more specific, such as the names of individual artists, album titles, or particular genres or subgenres—for example, “synth-pop”, “Britney Spears”, or “Grammy Award” are less likely to appear frequently throughout the entire corpus, but are crucial for differentiating relevant passages. These terms provide context directly related to the specific content of certain passages, thereby enhancing the ability of the retrieval system to match queries with the most relevant documents.

The primary advantage of this frequency-based token deletion approach lies in its ability to evaluate the relative importance of tokens based on their frequency of occurrence within the context of the entire corpus rather than relying solely on a predefined list of stopwords. This method adapts to the characteristics of an individual dataset dynamically, thereby enabling flexible applications in various domains. For instance, in a legal corpus, common terms such as “law”, “case” or “court” may appear frequently and contribute little to distinguishing specific passages, whereas terms related to particular legal cases, statutes, or specialized terminology would be far more informative. By removing frequently occurring tokens, the proposed method aims to minimize the influence of non-informative terms, thereby enabling the retrieval model to prioritize more meaningful ones. This selective deletion process improves the quality of the augmented data

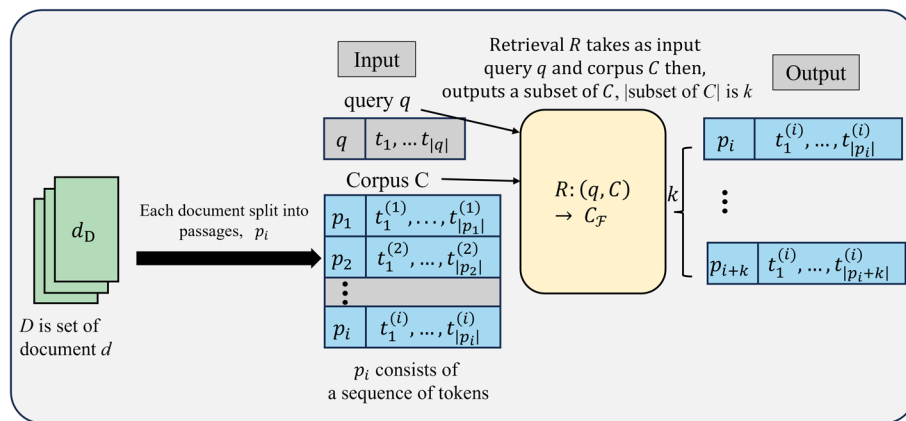


Fig. 10 The overall process of DPR. A document collection D is transformed into a passage corpus and, along with a query, is provided as input to the retrieval system. The system outputs the passage most relevant to the query. During this process, DPR trains encoders for both the passage corpus and the query

Table 4 Symbols and their explanations

Symbol	Name of symbol	Explain
D	Documents	A collection of documents
M	Number of passages	Number of passages originated from documents
C	Corpus	A set of passages
p_i	i th passage	A passage consists of a sequence of tokens
t	token	A token from the passage
q	query	A query consists of a sequence of tokens
R	Retrieval function	Retrieval function takes an input query and Corpus, and outputs a subset of the Corpus
C_F	Filtered Corpus	A subset of the Corpus

by preserving tokens that are likely to the accurate identification of relevant passages. Consequently, the overall performance of information retrieval systems is enhanced by emphasizing distinctive and contextually important information.

Training dense passage retrieval

Figure 10 illustrates the entire DPR workflow, depicting each step of the retrieval process. This visualization illustrates how the model is trained to retrieve relevant passages effectively. Table 4 presents the symbols used to describe the DPR training process, along with their corresponding explanations.

D denotes Document, which represents a collection of documents. In information retrieval, a dataset typically consists of multiple documents, where each document serves as a distinct unit of text that can be searched or analyzed for specific information. M denotes the number of passages extracted from the documents. In several information retrieval tasks, documents are divided into smaller segments or passages, each of which can be considered independently during the search process. C represents the corpus, which is the entire collection of passages considered in the retrieval system. This includes all passages extracted from various documents, forming the complete text dataset, with p_i denoting the i -th passage, where i indicates the passage index within the corpus. Each passage

consists of a sequence of tokens, which represent smaller units of text, such as words or subwords, denoted by t . In natural language processing, text is often tokenized into smaller components for analysis and processing. The query symbol q denotes a sequence of tokens input by the user, which can comprise words, phrases, or sentences. In information retrieval systems, a query is used to search for relevant passages or documents that match the input criteria. R denotes a retrieval function that takes a query and corpus as inputs and returns a subset of the corpus that is deemed relevant to the query. This function evaluates the similarity between the query and passages to identify the relevant results. C_f denotes the filtered corpus, i.e., a subset of the original corpus, selected based on certain criteria. For instance, it can consist of passages identified as relevant by a retrieval function. These symbols encapsulate essential concepts in information retrieval, providing a framework for understanding how a retrieval system processes documents and queries to deliver relevant results.

The corpus is segmented into passages of equal length to form a corpus comprising M passages. This is denoted by the corpus $C = \{p_1, p_2, \dots, p_M\}$. Each passage comprises tokens t .

$$p_i = \langle t_1^{(i)}, \dots, t_{|p_i|}^{(i)} \rangle$$

The information retrieval task is formulated as follows:

$$R : (q, C) \rightarrow C_f$$

Given an input question q , the retrieval task returns a filtered set C_f from the corpus C . Specifically, the cardinality of the filtered set is denoted by k , which is smaller than the size of the corpus.

$$|C_f| = k \ll |C|$$

Here, k is interpreted to represent the number of top-ranked items retrieved. The accuracy of the top- k documents returned by the model is used as a performance metric for the retrieval model.

DPR employs a dual BERT encoder-based structure [2]. The encoders encode input questions and passages independently. The similarity between the two encoded vectors is calculated, and the model is trained to maximize the similarity for correct question-passage pairs and minimize it otherwise. The function $E(\cdot)$ represents the encoding process that converts the inputs into vectors. Cosine similarity is used to calculate the similarity between the two input vectors. The learning objective function is defined as follows [2]:

$$L = -\log \frac{e^{\text{sim}(E(q), E(p_i^+))}}{e^{\text{sim}(E(q), E(p_i^+))} + \sum_{j=1}^n e^{\text{sim}(E(q), E(p_j^-))}}$$

With this objective, (q, p_i^+) represents a question and its corresponding i -th positive passage, whereas p_i^- represents an irrelevant passages. The goal is to ensure that the relevant passage p_i^+ exhibits a higher similarity score with the question q , compared to an irrelevant passage p_i^- .

Data augmentation algorithm

Algorithm 1 Data augmentation considering importance of tokens

Require: Passage dataset $C = \{p_1, p_2, \dots, p_m\}$, Question dataset $Q = \{q_1, q_2, \dots, q_{|Q|}\}$,
 Tokenizer $\mathcal{T}_{BERT}$, counter

Ensure: Answer passages for given queries $C' = \{p'_1, p'_2, \dots, p'_{|C'|}\}$

```

1: Initialize dictionary  $cf\_count$ ,  $T_i$  * Initialize count*
2: for  $i = 1$  to  $|C|$  do * Iterate over each passage*
3:   List of tokens:  $T_i \leftarrow \mathcal{T}_{BERT}.tokenize(p_i)$  * Tokenize the current passage*
4:   for Token  $t$  in  $T_i$  do * Consider each token*
5:     if  $t$  in  $cf\_count$  then * If seen token*
6:        $cf\_count[t] \leftarrow cf\_count[t] + 1$  * Add corpus frequency*
7:     else * If unseen token*
8:        $cf\_count[t] = 1$  * Assign the value one*
9:     end if * End of condition*
10:  end for * End of loop*
11:  for  $j = 1$  to  $|C|$  do * Iterate over each passage*
12:     $weight \leftarrow GM(T_i.counter(t), cf\_count(t))$  * Calculate weight*
13:     $normalized\ weight \leftarrow \frac{(weight\ of\ t - \min\ weight\ of\ T_i)}{(\max\ weight\ of\ T_i - \min\ weight\ of\ T_i) + \epsilon}$  * Normalize*
14:     $probability \leftarrow normalized\ weight \times \text{scaling factor}$  * Deletion probability*
15:     $C' \leftarrow \text{delete each token with acquired probability}$  * Augment the dataset*
16:  end for * End of loop*
17: end for * End of loop*
18: return  $C'$  * Return output*
```

The proposed data augmentation algorithm is designed to delete tokens from a passage selectively by evaluating their importance. Its primary objective is to enhance data augmentation by retaining tokens that convey significant meaning while removing those that are less relevant, allowing the model to concentrate on the essential content of each passage. The proposed data augmentation algorithm (Algorithm 1) enhances passage discriminability by deleting statistically redundant tokens based on their local and global importance. The method consists of the following steps:

Step 1: Input and Tokenization. The algorithm receives a passage dataset $C = \{p_1, p_2, \dots, p_m\}$, a question dataset $Q = \{q_1, q_2, \dots, q_{|Q|}\}$, a tokenizer $\mathcal{T}_{BERT}$, and a frequency counter. Each passage p_i is tokenized into a list of subword units T_i using $\mathcal{T}_{BERT}$. This step prepares the data for subsequent frequency-based analysis.

Step 2: Corpus Frequency (CF) Accumulation. For each token in the passage set, the algorithm updates a global dictionary cf_count that tracks corpus-level token frequencies. If a token has appeared before, its frequency is incremented; otherwise, it is initialized to one. This allows the model to measure how common each token is across the entire dataset.

Step 3: Passage Frequency (PF) and Weight Calculation. For each passage, the algorithm also calculates the passage frequency, which reflects how often a token appears within that specific passage. The importance of a token is then determined by computing the geometric mean of its CF and PF values:

$$\text{Weight}(t) = \sqrt{\text{CF}(t) \cdot \text{PF}(t)}$$

Tokens that are frequent both locally (within a passage) and globally (across passages) receive higher weights, indicating lower specificity.

Step 4: Weight Normalization and Deletion Probability. The token weights are normalized using min-max scaling:

$$\tilde{w}_t = \frac{w_t - \min(W)}{\max(W) - \min(W) + \epsilon},$$

where W is the set of weights within a passage and ϵ is a small constant to prevent division by zero. The normalized weights are multiplied by a scaling factor to compute the deletion probabilities.

Step 5: Token Deletion and Output Generation. Each token is deleted probabilistically based on its assigned deletion probability. This probabilistic filtering removes less informative tokens while retaining those that convey passage-specific meaning. The result is a new set of augmented passages C' that emphasizes discriminative content without relying on external augmentation sources. This selective deletion process ensures that the output retains the most critical information while introducing controlled variations for effective data augmentation. Furthermore, by maintaining alignment with question-answer pairs, the resulting passages enhance the generalizability of the model and improve its performance on unseen queries.

Experimental results

Experimental environment

The query and passage are transformed into vector representations for similarity calculation. An encoder is used to generate vector representations for both the query and the passage. Various types of neural networks can be used as encoders. Two independent BERT networks (bert-base-uncased) were employed in this experiment. The encoder outputs a representation of the {CLS} token. The dimensionality of the token is 768.

Building on the methodology of previous studies [41], the English Wikipedia dump is used as the source of documents for answering the questions. The preprocessing code [42] is applied to extract clean text sections from Wikipedia articles. This process effectively removes semi-structured data, including tables, info-boxes, lists, and disambiguation pages. Subsequently, each article is segmented into multiple disjoint text blocks of 100 words each, serving as basic retrieval units, following the technique described in a previous study [43]. This segmentation yields a total of 21,015,324 passages. In addition, each passage is prefixed with the title of the corresponding Wikipedia article and an {SEP} token. In this experiment, the WikiQA [44] and Natural Questions (NQ) [45] datasets are utilized. WikiQA is a dataset of question-answer pairs specifically designed for open-domain question-answering research. To better represent the real-world information requirements of general users, questions in WikiQA are sourced from Bing query logs. Each question is linked to a Wikipedia page likely to contain a relevant answer. A larger NQ dataset is also available for the end-to-end question-answering tasks. The questions in the NQ are derived from real Google search queries, with answers identified as specific spans within Wikipedia articles by the annotators. For Natural Questions, since the original passages are split and processed differently from the candidate passage pool, each gold passage containing the answer passage is aligned and replaced with the corresponding passage in the candidate pool. If no match is identified due to discrepancies in Wikipedia versions or preprocessing methods, the questions are discarded. WikiQA uses only pairs in which the questions and answers are matched accurately. Table 5 presents the dataset sizes and the number of questions in each split used for training and evaluation.

During the training process, a distributed learning approach is utilized based on multiple GPUs. The batch size is set to 16, and the question and passage encoders are trained

Table 5 Brief statistics of datasets employed in our empirical experiments

Dataset	Size	Train	Dev	Test
WikiQA (general QA)	6.8M	856	122	122
Natural Question (general QA)	42G	58,880	8,757	3,610

on the smaller WikiQA and larger NQ datasets, respectively. The learning rate is set to $2e-5$, using the Adam optimizer. In addition, linear scheduling with a warm-up phase is applied to manage the learning process. To evaluate the effects of data augmentation, training is conducted on three different datasets—the original dataset without augmentation, the dataset augmented with the traditional RD method, and the dataset augmented using the proposed method. Data augmentation is performed to double the size of the dataset. For the experiment without data augmentation, the original dataset is duplicated to ensure consistency in terms of dataset size. Data augmentation is applied exclusively to positive contexts, which serves as justification for the correct answer to each query. For random token deletion, each token is removed with a uniform probability of 0.1. For weighted token deletion, the weight of each token is calculated as the product of its PF and CF values, followed by min-max normalization of these weights. The normalized weights are then scaled to values within the range $[0, 0.2]$, with a median value of 0.1. The deletion probability for each token is determined based on these scaled weights, enabling selective deletion that reflects token importance.

Top-k Accuracy is commonly used to evaluate retrieval models because determining the exact answer within a corpus can often be a difficult task [46]. For instance, in DPR, the model returns multiple passages for a given query, and the evaluation determines whether the correct passage is included among the top-k retrieved results. This is particularly important in retrieval systems, where the ability of the model to include the correct answer within the top-ranked results is crucial. This metric offers a more comprehensive assessment of model performance than an evaluation based only on the single highest-ranked result [47]. The corresponding equation is as follows:

$$Top - k Accuracy = \frac{1}{N} \sum_{i=1}^N 1\{y_i \in \hat{y}_i^{(K)}\} \quad (4)$$

, where N represents the total number of test samples, k denotes the top-k candidate classes or passages predicted by the model, y_i denotes the true label of the i -th data point, $\hat{y}_i^{(K)}$ denotes the set of the top k classes or passages predicted by the model for the i -th data point, and $1\{\cdot\}$ is an indicator function that returns one if the condition is true and zero if it is false.

Comparison results

Data augmentation is performed using two methods: uniform random deletion with a probability of 0.1 and weighted deletion based on token importance. Table 6 provides an example from the dataset. For the query “who sang waiting for a girl like you”, which is related to a positive document, the token “Foreigner” is deleted under uniform random deletion. In contrast, the token “of” is deleted based on the PF and CF weights in

Table 6 The real example of augmented data

Query	Who sang waiting for a girl like you
Original passage	Waiting for a Girl Like You "Waiting for a Girl Like You" is a 1981 power ballad by the British-American rock band Foreigner. The distinctive synthesizer theme was performed by the then-little-known Thomas Dolby, and this song also marked a major departure from their earlier singles because their previous singles were mid to upper tempo rock songs while this song was a softer love song with the energy of a power ballad. It was the second single released from the album "4" (1981) and was co-written by Lou Gramm and Mick Jones. It has become one of the band's most.
Uniform probability random deletion	Waiting for a Girl Like You "Waiting for a Girl Like ___" is a 1981 power ballad by the British-American rock band ___. The distinctive synthesizer theme was performed by the then-little-known Thomas Dolby, and this song also marked a major departure from their earlier singles because their previous singles were mid to upper tempo ___ songs while this song was a softer love song with the energy of a power ballad. It was the second single released from the album "4" (1981) and was co-written by Lou Gramm and Mick Jones. It has become one of the band's most.
Proposed method	Waiting for a Girl Like You "Waiting for a Girl Like You" is ___ 1981 power ballad by the British-American rock band Foreigner. The distinctive synthesizer theme was performed by the then-little-known Thomas Dolby, and this ___ also marked a major departure from their earlier singles because their previous singles were mid to upper tempo rock songs while this song was a softer love song with the energy of a power ballad. It was the second single released from the album "4" (1981) and was co-written by Lou Gramm and Mick Jones. It has become one ___ band's.

The original passage was subjected to both the conventional Uniform Probability Random Deletion method and the proposed Weighted Deletion method. The conventional method randomly removes tokens, as seen with the deletion of "you" and "rock," as well as "Foreigner," a token that serves as a critical clue for answering the query. In contrast, the proposed method calculates the importance of tokens and deletes tokens with relatively lower importance, such as "a" or "song"

Table 7 Comparison of methods on Top-k accuracy for WikiQA and natural question

Method	Top-10		Top-20		Top-30		Top-40	
	WikiQA	NQ	WikiQA	NQ	WikiQA	NQ	WikiQA	NQ
Proposed method	<u>0.844</u>	<u>0.720</u>	<u>0.903</u>	<u>0.768</u>	<u>0.924</u>	<u>0.796</u>	<u>0.932</u>	<u>0.810</u>
DPR (base)	0.594	0.712	0.691	0.762	0.737	0.789	0.766	0.803
Uniform probability random deletion	0.821	0.708	0.866	0.761	0.908	0.786	0.921	0.802

The proposed method consistently yielded better results for both the small-scale WikiQA dataset and the large-scale Natural Question (NQ) dataset across Top-k@ (10, 20, 30, 40) evaluations, with the improvements highlighted in bold. Notably, the performance increase was particularly pronounced for the small-scale dataset, and this significant improvement is underlined for emphasis

weighted token deletion. "Foreigner" is the name of the band who sang the queried song and a critical token for answering the query about who performed the song. In contrast, the token "of" carries minimal significance. This example illustrates the effectiveness of the proposed method in augmenting data by selectively removing non-essential tokens.

Table 7 presents the top-k accuracy results for different data augmentation methods on datasets of various sizes. The proposed method, which employs CF and PF-based weighted deletion, exhibits particularly high-quality performance on the small-scale WikiQA dataset. The initial hypothesis is that the proposed method offers more significant benefits on smaller datasets, particularly those with a limited number of question-answer pairs. Table 7 indicates that the proposed corpus-passages frequency-based data augmentation method achieves significant performance improvements on the WikiQA dataset, supporting our initial hypothesis. The numerical results reveal clear and significant improvements, confirming that the proposed method effectively preserves key

information while enhancing the model performance in cases where the dataset size is limited. In terms of the Top-10 results, the baseline DPR model without augmentation exhibits an accuracy of 0.594. When trained with uniform random deletion, the accuracy increases to 0.821. The model trained with the augmented data using the proposed weighted deletion method achieves the highest accuracy of 0.844. Similar trends are observed for all values of k . For smaller datasets, the model trained on the augmented data is consistently observed to outperform the model trained without augmentation. This emphasizes the effectiveness of data augmentation using a deletion-based strategy. In addition, the model trained with the augmented data generated using the proposed deletion-based method is observed to perform the best. These findings confirmed that the proposed method is superior to traditional data augmentation techniques.

Besides the significant improvements on the small-scale WikiQA dataset, the proposed method also exhibits competitive performance on the larger NQ dataset. In terms of Top-40 results, the accuracy of the baseline DPR model without augmentation is 0.803 on this dataset. The model trained with uniform random deletion achieves an accuracy of 0.802, indicating little improvement over the baseline. These results indicate that the use of the traditional uniform random deletion method for data augmentation, in this case, has minimal impact on performance. However, the model trained with the proposed weighted data augmentation method consistently achieves higher top- k accuracies corresponding to various k values. Considering the inherent challenge of achieving percentage-point improvements in information retrieval tasks, the stability and improved accuracy demonstrate the effectiveness of the proposed approach. These findings highlight that the proposed method is not only effective on small datasets but also competitive on larger ones. The consistent improvements across various top- k values in both cases demonstrate the effectiveness of the proposed approach, establishing it as a superior data augmentation strategy for information retrieval tasks.

Our experiments also include a comparative analysis of a data augmentation approach in which {MASK} tokens are inserted into the positions of the tokens deleted using the proposed method. Table 8 shows that augmenting the data by inserting {MASK} tokens into the positions of deleted tokens yields superior performance compared to only using the deletion-based method. This improvement is attributed to the underlying architecture of BERT, which serves as the backbone of the DPR architecture. During pre-training, BERT employs a self-attention mechanism to capture the relationships between tokens across the entire input sequence. During this process, the {MASK} token provides an explicit signal to the model, guiding it to predict the information corresponding to the masked position. Thus, by inserting {MASK} tokens, the model is implicitly

Table 8 The difference in performance of the proposed method on WikiQA is based on the presence or absence of {Mask} token insertion

Method	Top-10	Top-20	Top-30	Top-40
Weighted-based deletion	0.844	0.903	0.924	0.932
Insertion of mask tokens in place of weighted-based deletion	0.860	0.908	0.926	0.941

The experimental results showed that inserting a {MASK} token in the deleted positions demonstrated better outcomes, with these values highlighted in bold

notified of the removed tokens, encouraging it to reconstruct the missing information based on the surrounding context during the passage retrieval training process.

In contrast, the weight-based token deletion strategy focuses only on removing tokens of low importance. Although effective, this method lacks additional signals required to enhance the capacity of the model to predict and reconstruct missing information. However, inserting {MASK} tokens after a weight-based deletion improves the predictive capacity of the model by introducing a restoration task. This not only helps maintain the contextual integrity of the input but also enables the model to learn more nuanced relationships within the data. These findings indicate that combining weight-based deletion with {MASK} token insertion provides a dual advantage and maintains the benefits of removing less informative tokens while challenging the model to reconstruct meaningful representations. This process likely enhances the robustness and generalizability of the model, particularly in tasks such as passage retrieval, where understanding the context and reconstructing missing elements are crucial. Consequently, this approach exhibits significant potential for further improving the performance of question-answering and information-retrieval systems.

To further investigate the impact of augmentation multiples on retrieval performance, we conducted additional experiments by varying the augmentation ratio from 2x to 6x using the proposed method. Table 9 summarizes the Top- k accuracy results on the WikiQA dataset. Here, Nx denotes that each original passage is augmented into N versions, effectively increasing the dataset size by a factor of N . The results show that retrieval performance improves steadily from 2x to 4x augmentation, with the Top-40 accuracy increasing from 0.932 to 0.972. However, beyond 4x, performance gains diminish and even begin to degrade. For instance, with 6x augmentation, Top-40 accuracy drops to 0.866. This trend suggests that while moderate augmentation is beneficial for enhancing model generalization, excessive augmentation may introduce redundancy or noise that hinders learning. These findings indicate that the proposed augmentation method is most effective within a practical range (2x to 4x), where it significantly enhances model performance on small-scale datasets without relying on external data.

Table 10 presents a combined analysis of different deletion strategies and augmentation targets. The results demonstrate that the proposed method, when applied solely to positive passages, yields the highest performance across all top- k accuracy metrics. This highlights the effectiveness of our frequency-based token deletion approach in enhancing the diversity and quality of positive examples. In addition, applying the same augmentation to both positive and negative passages resulted in marginal improvements

Table 9 Top- k accuracy on WikiQA based on data augmentation multiples of the proposed method

Method	Top-10	Top-20	Top-30	Top-40
Base (No Augmentation)	0.594	0.691	0.737	0.766
2x	0.844	0.903	0.924	0.932
3x	0.929	0.951	0.967	0.969
4x	0.942	0.967	0.972	0.972
5x	0.891	0.915	0.922	0.931
6x	0.824	0.841	0.850	0.866

The results demonstrate that higher data augmentation ratios lead to overall improved performance

Table 10 Comparison of different deletion strategies and augmentation targets on the WikiQA dataset

Method/Setting	Top-10	Top-20	Top-30	Top-40
Proposed Method (positive-only augmentation)	0.844	0.903	0.924	0.932
Proposed Method (positive + negative augmentation)	0.831	0.889	0.909	0.922
Stopword-only Deletion	0.784	0.857	0.888	0.905

The results highlight the effectiveness of the proposed method and the impact of augmenting positive vs. both positive and negative passages

Table 11 Ablation study on CF/PF weighting and deletion weight scaling

Setting	Top-10	Top-20	Top-30	Top-40
Proposed method	0.844	0.903	0.924	0.932
CF/PF weighting ($\alpha, \beta \times 1.5$)	0.826	0.873	0.901	0.916
Deletion weight scaling ($w \times 2$)	0.831	0.886	0.910	0.922

The proposed setting refers to the default configuration used in our main experiments

over the uniform deletion baseline but slightly underperformed compared to positive-only augmentation. These findings suggest that while augmenting negative examples introduces minor complementary effects, the primary gains are derived from enriched positive-side supervision. Stopword-only deletion consistently underperformed compared to our method and even the uniform deletion baseline, reinforcing the importance of dynamically selecting tokens for deletion based on corpus- and passage-level statistics rather than relying on static stopwords lists.

To assess the robustness of the proposed method with respect to key hyperparameters, we conducted ablation studies focusing on the weighting ratio between CF and PF, as well as the scaling applied to the computed deletion weights. The results are summarized in Table 11. When increasing the contribution of CF and PF in the geometric mean computation by scaling both α and β by a factor of 1.5, we observed a noticeable performance drop across all Top- k metrics. Similarly, doubling the deletion weight values before normalization led to a slight degradation in retrieval performance. These results suggest that while the method is generally robust, overly aggressive weighting or scaling may reduce its effectiveness by either misrepresenting token importance or skewing the probability distribution. Overall, the proposed configuration achieved the highest retrieval accuracy, confirming the effectiveness and empirical stability of the default hyperparameter settings used in our main experiments.

Discussion

This study demonstrates the potential of the proposed corpus-passage frequency-based token deletion technique for improving DPR performance. A notable observation is that the proposed method exhibits significantly stronger effects on small-scale datasets, particularly in addressing class imbalances through data augmentation. The experimental results reveal that using the proposed method improves Top-10 accuracy on the WikiQA dataset by approximately 2.3% compared to the traditional random deletion method.

This highlights that the frequency-based token deletion approach enhances the quality of augmented data while minimizing information loss, making it especially effective in scenarios involving limited question-passage pairs.

This study investigates the impact of combining {MASK} token insertion with the frequency-based deletion method. The experiments reveal that inserting {MASK} tokens preserves the passage context while guiding the model to recover the missing information, thereby positively influencing the retrieval performance. On the WikiQA dataset, the Top-10 accuracy increases by 1.6% when {MASK} tokens are applied. This improvement can be attributed to the BERT architecture, which is trained to leverage {MASK} tokens during pretraining to capture deeper contextual relationships. These results highlight the utility of complementary techniques, such as MASK token insertion, in improving the ability of the model to reconstruct and understand contexts during passage retrieval tasks.

Finally, an analysis of the data augmentation ratios reveals that increasing the augmentation significantly enhances the generalization performance of the DPR. When the augmentation ratio is increased fourfold, the Top-10 accuracy on the WikiQA dataset increases by over 15%. This demonstrates the effectiveness of data augmentation in low-resource environments, where expanded datasets introduce greater diversity into the training process, thereby enhancing model performance. These findings provide strong evidence that higher augmentation ratios can be beneficial on small datasets and open new possibilities for broader applications in various natural language processing tasks.

Conclusion

This study introduces a novel token deletion method based on corpus-passage frequency to address the limitations of existing data augmentation techniques in DPR tasks. Unlike traditional methods, such as random token deletion, which often results in the loss of critical contextual information, the proposed approach selectively removes less important tokens based on their frequency of appearance within individual passages and throughout the corpus. This selective strategy ensures the preservation of key information while enhancing the ability of the model to generalize and retrieve more accurate results. The effectiveness of this approach is particularly evident in notable improvements on smaller datasets such as WikiQA, highlighting its suitability for environments with limited data resources. While our method maintains stable performance on larger datasets such as NQ, the relative improvement observed in large-scale settings is comparatively modest. These observations highlight that our frequency-based selective deletion method is particularly advantageous for low-resource scenarios where effective data augmentation can substantially enhance the model's generalization capabilities. By assigning greater importance to contextually significant tokens and systematically removing less relevant tokens, the proposed method enhances the ability of the retrieval model to differentiate between relevant and irrelevant passages. This leads to an improved generalizability and a more refined learning process, particularly in open-domain question-answering and other retrieval-based tasks.

In addition to its effectiveness, the proposed method addresses key limitations of previous augmentation techniques, such as EDA, which often introduce noise or result in the loss of meaningful information. By selectively preserving domain-specific language

and specialized terminology, the frequency-based approach is demonstrated to be particularly suitable for tasks that require precise contextual understanding. Further, its heightened effectiveness is confirmed experimentally on smaller datasets, reinforcing its general applicability and potential for handling diverse types of data. Beyond DPR, this method holds significant promise for broader applications, including natural language processing tasks, such as chatbots, automatic translation, and specialized question-answering systems. By enhancing the augmentation process, the proposed approach facilitates improved generalizability across tasks, thereby providing practical benefits in real-world scenarios. This lays a strong foundation for future developments in data augmentation techniques and opens up new possibilities for advancing the field of natural language processing.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2021-II211341, Artificial Intelligence Graduate School Program (Chung-Ang University)).

Author contributions

A.M. reviewed the manuscript, evaluated the experimental results, and contributed to the overall editing process. K.K. implemented the proposed method and conducted the experimental analysis. J.L. supervised the research, provided conceptual guidance, and contributed to manuscript revisions. All authors reviewed and approved the final manuscript.

Funding

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2021-II211341, Artificial Intelligence Graduate School Program (Chung-Ang University)).

Data availability

Data supporting the findings of this study are available in <https://www.microsoft.com/en-us/download/details.aspx?id=52419> and <https://ai.google.com/research/NaturalQuestions>

Declarations

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Competing interests

The authors declare no Competing interests.

Received: 6 January 2025 Accepted: 3 August 2025

Published online: 13 August 2025

References

- Robertson S, Zaragoza H. The probabilistic relevance framework: Bm25 and beyond. *Found Trends Inf Retr*. 2009;3(4):333–89.
- Karpukhin V, Oguz B, Min S, Lewis P, Wu L, Edunov S, Chen D, Yih W-T. Dense passage retrieval for open-domain question answering. In: *Proceedings of the 15th Conference on Empirical Methods in Natural Language Processing, Virtual*; 2020. p. 6769–6781.
- He H, Garcia EA. Learning from imbalanced data. *IEEE Trans Knowl Data Eng*. 2009;21(9):1263–84.
- Gao L, Dai Z, Callan J. COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Mexico City, Mexico*; 2021. p. 3030–3042.
- Formal T, Piwowarski B, Clinchant S. Splade: Sparse lexical and expansion model for first stage ranking. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual* 2021.
- Luan Y, Eisenstein J, Toutanova K, Collins M. Sparse, dense, and attentional representations for text retrieval. *Trans Assoc Comput Linguist*. 2021;9(1):329–45.
- Reichman B, Heck L. Dense passage retrieval: Is it retrieving? In: Al-Onaizan, Y., Bansal, M., Chen, Y.-N. (eds). *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, USA; November 12–16, 2024*. p. 13540–13553.

8. Dong Z, Ni J, Bikel D, Alfonseca E, Wang Y, Qu C, Zitouni I. Exploring dual encoder architectures for question answering. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates; 2022. p. 9414–9419.
9. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 17th Conference of the North American Chapter of the Association for Computational Linguistics, Minnesota, USA; June 2–7, 2019. p. 4171–4186.
10. Song X, Salcianu A, Song Y, Dopson D, Zhou D. Fast WordPiece tokenization. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Repub; 2021. p. 2089–2103.
11. Tenney I, Das D, Pavlick E. BERT rediscovers the classical NLP pipeline. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy; 2019. p. 4593–4601.
12. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA; 2017. p. 6000–6010.
13. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA; 2016. p. 770–778.
14. Clark K, Khandelwal U, Levy O, Manning CD. What does BERT look at? an analysis of BERT's attention. In: Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Florence, Italy; 2019. p. 276–286.
15. Ren R, Lv S, Qu Y, Liu J, Zhao WX, She Q, Wu H, Wang H, Wen J-R. PAIR: leveraging passage-centric similarity relation for improving dense passage retrieval. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Bangkok, Thailand; August 1–6, 2021. p. 2173–2183.
16. Wang L, Yang N, Huang X, Jiao B, Yang L, Jiang D, Majumder R, Wei F. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, Canada; 2023. p. 2244–2258.
17. Musmann S, Ermon S. Learning and inference via maximum inner product search. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning, New York, USA; 2016. p. 2587–2596.
18. Malkov YA, Yashunin DA. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans Pattern Anal Mach Intell*. 2020;42(4):824–36.
19. Guu K, Lee K, Tung Z, Pasupat P, Chang M. Retrieval augmented language model pre-training. In: Proceedings of the 37th International Conference on Machine Learning, Virtual; 2020. p. 3929–3938.
20. Zhou K, Gong Y, Liu X, Zhao WX, Shen Y, Dong A, Lu J, Majumder R, Wen J-r, Duan N. SimANS: Simple ambiguous negatives sampling for dense text retrieval. In: Proceedings of the 17th Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates; 2022. p. 548–559.
21. Sachan DS, Lewis M, Yogatama D, Zettlemoyer L, Pineau J, Zaheer M. Questions are all you need to train a dense passage retriever. *Trans Assoc Comput Linguist*. 2023;11(1):600–16.
22. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. *J Big Data*. 2019;6(60):1–48.
23. Liu P, Wang, X., Xiang, C., Meng, W.: A survey of text data augmentation. In: 2020 International Conference on Computer Communication and Network Security (CCNS), Xi an, China; 2020. p. 191–195.
24. Feng SY, Gangal V, Wei J, Chandar S, Vosoughi S, Mitamura T, Hovy E. A survey of data augmentation approaches for NLP. In: Findings of The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Bangkok, Thailand; 2021. p. 968–988.
25. Wei J, Zou K. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In: Proceedings of the 14th Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China; 2019. p. 6382–6388.
26. Shorten C, Khoshgoftaar TM, Furht B. Text data augmentation for deep learning. *J Big Data*. 2021;8(1):101–35.
27. Kim H-S, Kang Y, Lee J-H. STAGE: Simple text data augmentation by graph exploration. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italy; 2024. p. 15238–15256.
28. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D. Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, Canada; 2020. p. 1877–1901.
29. Madsen A, Meade N, Adlakha V, Reddy S. Evaluating the faithfulness of importance measures in NLP by recursively masking allegedly important tokens and retraining. In: Findings of the 17th Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates; 2022. p. 1731–1751.
30. Chalkidis I, Fergadiotis E, Malakasiotis P, Androutsopoulos I. Large-scale multi-label text classification on EU legislation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy; 2019. p. 6314–6322.
31. Jin Q, Dingra B, Liu Z, Cohen W, Lu X. PubMedQA: a dataset for biomedical research question answering. In: Proceedings of the 14th Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China; 2019. p. 2567–2577.
32. Chalkidis I, Androutsopoulos I, Aletras N. Neural legal judgment prediction in English. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy; 2019. p. 4317–4323.
33. Jin Y, Jang E, Cui J, Chung J-W, Lee Y, Shin S. DarkBERT: A language model for the dark side of the Internet. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, Canada; 2023. p. 7515–7533.
34. Karimi A, Rossi L, Prati A. AEDA: An easier data augmentation technique for text classification. In: Findings of the 16th Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic; 2021. p. 2748–2754.

35. Qu Y, Ding Y, Liu J, Liu K, Ren R, Zhao WX, Dong D, Wu H, Wang H. Rocketqa: an optimized training approach to dense passage retrieval for open-domain question answering. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT; June 6–11, 2021. p. 5835–5847. <https://doi.org/10.18653/v1/2021.NAACL-MAIN.466>. Accessed 11 Aug 2025.
36. Ge S, Xiong C, Rosset C, Overwijk A, Han J, Bennett P. Augmenting zero-shot dense retrievers with plug-in mixture-of-memories. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore; 2023. p. 1796–1812. <https://doi.org/10.18653/v1/2023.emnlp-main.111>. <https://aclanthology.org/2023.emnlp-main.111/>. Accessed 11 Aug 2025.
37. Chen H, Dou Z, Mao K, Liu J, Zhao Z. Generalizing conversational dense retrieval via LLM-cognition data augmentation. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand; 2024. p. 2700–2718. <https://doi.org/10.18653/v1/2024.acl-long.149>. Accessed August 11, 2025. <https://aclanthology.org/2024.acl-long.149/>. Accessed 11 Aug 2025.
38. Li X, Li X, Zhang H, Du Z, Jia P, Wang Y, Zhao X, Guo H, Tang R. Syneg: Llm-driven synthetic hard-negatives for dense retrieval. arXiv preprint [arXiv:2412.17250](https://arxiv.org/abs/2412.17250) 2024.
39. Sennrich R, Haddow B, Birch A. Improving neural machine translation models with monolingual data. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany; August 7–12, 2016. 86–96.
40. Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning, Virtual; 2020. p. 1597–1607.
41. Lee K, Chang M-W, Toutanova K. Latent retrieval for weakly supervised open domain question answering. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy; 2019. p. 6086–6096.
42. Chen D, Fisch A, Weston J, Bordes A. Reading Wikipedia to answer open-domain questions. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada; 2017. p. 1870–1879.
43. Wang Z, Ng P, Ma X, Nallapati R, Xiang B. Multi-passage BERT: a globally normalized BERT model for open-domain question answering. In: Proceedings of the 14th Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China; 2019. p. 5878–5882.
44. Yang Y, Yih W-t, Meek C. WikiQA: A challenge dataset for open-domain question answering. In: Màrquez, L., Callison-Burch, C., Su, J. (eds). Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal; 2015. p. 2013–2018.
45. Kwiatkowski T, Palomaki J, Redfield O, Collins M, Parikh A, Alberti C, Epstein D, Polosukhin I, Devlin J, Lee K, Toutanova K, Jones L, Kelcey M, Chang M-W, Dai AM, Uszkoreit J, Le Q, Petrov S. Natural questions: a benchmark for question answering research. *Trans Assoc Comput Linguist*. 2019;7(1):452–66.
46. Siriwardhana S, Weerasekera R, Wen E, Kaluarachchi T, Rana R, Nanayakkara S. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Trans Assoc Comput Linguist*. 2023;11(1):1–17.
47. Voorhees EM. The TREC question answering track. *Nat Lang Eng*. 2001;7(4):361–78.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.