Retrieval-Augmented Generation in a Low Lexical Diversity Scenario

Michele Panteleo
Politecnico di Torino
s308511@studenti.polito.it

Prima Acharjee
Politecnico di Torino
s329198@studenti.polito.it

Minal Jamshed
Politecnico di Torino
s329091@studenti.polito.it

Abstract

This research focuses on the application of Retrieval-Augmented Generation (RAG) in a low-lexical-diversity setting, with a focus on classifying gas pipe damage descriptions for patch applicability. To quantify lexical similarities between documents, we introduce a new metric called lexical entropy, while to address class imbalance, we propose [+]EXPL, a data re-balancing technique designed to enhance the retrieval of patchable [+] cases. K-Nearest-Neighbors guide our selection of Sentence-BERT-NLI (SBERT-NLI) as the preferred encoder, as it proves more effective than state-of-the-art models in Semantic Search by capturing crucial nuanced differences between documents. Our results show that Mistral-7B exhibits some understanding of gas leak patchability with an F1-Macro score of 0.68, which improves to 0.87 when supported by the retrieval system. Given the synthetic nature of the documents, we avoid fine-tuning due to concerns that such an approach may not generalize well to human-written descriptions, where vocabulary variety would likely be higher. Nonetheless, RAG-driven approaches show promise in semantically constrained environments.

1. Introduction

The objective of this study is to evaluate the potential of a Retrieval-Augmented Generation (RAG) system in developing a chatbot designed to assist gas fitters in the application of Patch MadFlex [2], a product of Composite Research (CoRe). As we examined the corpus provided for this task, we identified that its structure—consisting of highly similar damage descriptions with minimal lexical variation—presents a distinct challenge. This low lexical diversity setting shapes both the retrieval and classification processes, making it a central focus of our study.

The dataset consists of synthetically generated descriptions of gas leaks, derived from a tabular dataset where structured boolean and categorical features have been transformed into textual descriptions. This process results in documents that are highly similar, with only slight variations in wordings. Consequently, two primary challenges arise: (1) a severe class imbalance, with only 1% damage descriptions labeled as *patchable* (or repairable, positive, *[+]*), and (2) low lexical diversity; which complicates both classification and retrieval tasks.

Given these challenges, we devised an approach that combines several key elements to effectively address them. First, we introduce a new metric, lexical entropy, to assess the degree of lexical similarity within the corpus. By comparing the lexical entropy of our dataset to that of other general-purpose and domain-specific corpora, we show that our documents exhibit greater lexical similarity to one another, sharing a more constrained set of terms. This observation sparked the intuition that conventional semantic search models may struggle to capture the subtle distinctions that are essential to this task (two damage cases may be nearly identical, but a single characteristic — such as the pressure level — can determine whether a leak is patchable or not). This led us to hypothesize that models that focus more on logical relationships, rather than purely semantic ones, would be more effective.

In line with this, we tested different encoder models for the retrieval component, ultimately finding that SBERT-NLI [6] (named SBERT in this work) outperformed other state-of-the-art models, such as MPNET [10], confirming the importance of natural language inference (NLI) capabilities in this domain. For the generative component, we selected Mistral7B [3] after evaluating it alongside other models, such as Llama3.2 and Llama2-13B, in a zero-shot setting. This allowed us to isolate the contribution of the retrieval system before assessing the generative performance. Mistral7B demonstrated strong in-domain knowledge and performed particularly well when the few-shot setting provided a balanced mix of positive and negative examples. However, given the dominance of negative examples in our dataset, Mistral7B was initially overwhelmed by these, despite the good retrieval performance of SBERT. To address this imbalance, we introduce a technique called [+]EXPL, which effectively cleans up positive neighborhoods by removing negatives that are too close to positive instances. This ensures that Mistral7B can focus more on the positive cases, leading to a substantial improvement in performance when integrated into the final RAG system.

This work is entirely fine-tuning free. Overfitting to the limited vocabulary of our corpus is a serious risk, potentially leading to misleading evaluations. In real-world applications, where the system relies on human-written descriptions with a broader vocabulary, this could undermine per-

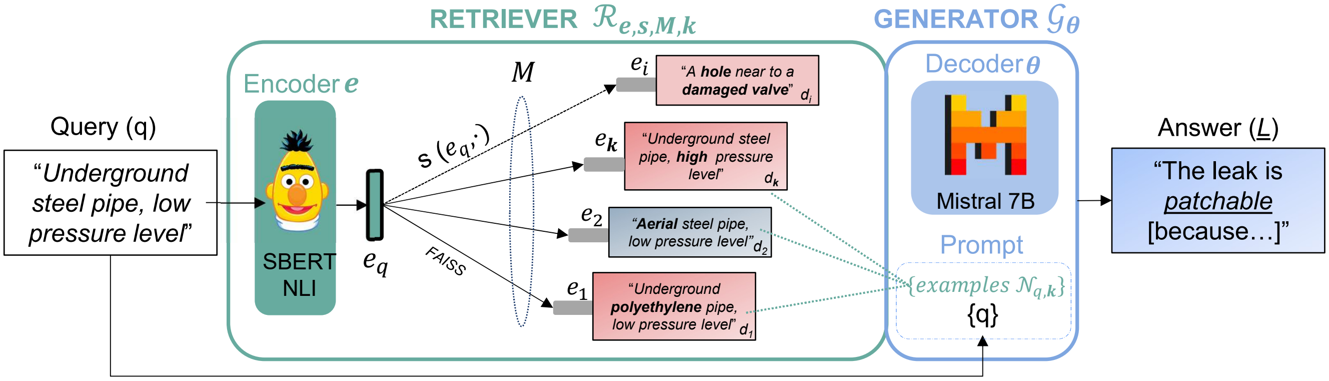
RETRIEVER $\mathcal{R}_{e,s,M,k}$
GENERATOR $\mathcal{G}_\theta$
Query (q)
"Underground steel pipe, low pressure level"
Encoder e
SBERT NLI
e_q
$s(e_q, \cdot)$
FAISS
M
e_i
"A hole near to a damaged valve" d_i
e_k
"Underground steel pipe, high pressure level" d_k
e_2
"Aerial steel pipe, low pressure level" d_2
e_1
"Underground polyethylene pipe, low pressure level" d_1
Decoder θ
Mistral 7B
Prompt
{examples $\mathcal{N}_{q,k}$}
{q}
Answer (L)
"The leak is patchable [because…]"

Figure 1. **Our RAG System**. Query is encoded by SBERT, then FAISS is leveraged for fast retrieval of the k most similar cases based on the dot product (MIPS). The retrieved top-k examples complete the prompt, which is then processed by Mistral7B to perform the task. In our experimental setting, we found the best results achieved with k = 9. The prompt constrains the model to only label the case, streamlining evaluation. The explanation (*because...*) is not investigated in this work.

formance.

This paper is organized as follows: §2 reviews related work in the fields of semantic search and retrieval-augmented generation. §3 presents the methodology, detailing the lexical entropy metric, encoder and decoder selection, the [+]EXPL technique, and the downsampling approach, whose results are reported in §4. Finally, §5 sets the final considerations and introduces ideas for future works.

2. Related Works

Retrieval-Augmented Generation (RAG) has emerged as a promising approach to improve knowledge-intensive Natural Language Processing tasks by integrating retrieval mechanisms with generative models [5]. Our study applies this approach to a domain-specific corpus of gas pipe damage descriptions, utilizing pre-trained LLMs for retrieval and classification. Reinhart et al. (2024) [7] evaluated the linguistic styles of LLMs, revealing significant differences in complex grammatical usage compared to human writing. Our project investigates the lexical entropy of the corpus, focusing on the generation of synthetic descriptions for gas pipe repairs by LLMs and assessing how their rigidity impacts the corpus lexical variety. Although outdated, SBERT-NLI [6] emerged as the most effective retrieval mechanism, while excelling models such as MPNet [10] tend to focus on other characteristics of a document. SBERT focused capabilities in semantic similarity evaluation are useful for perceiving subtle variations and contradictions in the text. In contrast, MPNet strength in analyzing contextual relationships did not prove beneficial in understanding the nuanced use of language in a domain-specific corpus like ours. Furthermore, LLMs require carefully crafted prompt engineering to effectively extract intrinsic knowledge and improve accuracy in classification tasks [1]. Our project explores the integration of LLMs to generate answers for patch applicability in gas pipe repairs and to leverage prompt engineering to refine input structures, significantly improving accuracy and utility.

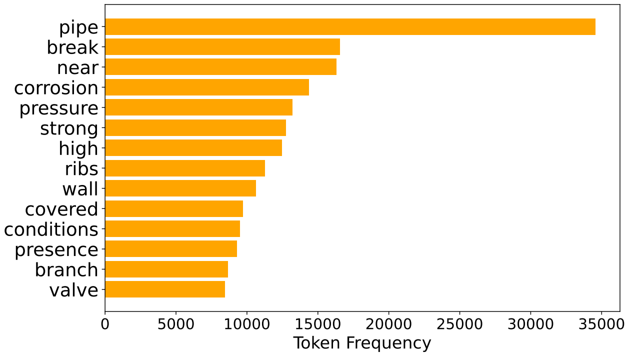

Figure 2. **Most frequent words in Gas Pipe Repairs corpus** excluding stop words and punctuations.

3. Methodology

In this section, we outline the procedure that led to the development of the Retrieval-Augmented Generation System (RAGS) depicted in 1. We begin by defining the Corpus Lexical Entropy, which defines a measure to quantify the extent of document similarity. Thereafter, we provide the mathematical formulation of our problem to set the scene for the upcoming parts of this section.

3.1. Corpus Lexical Entropy

We define the Lexical Entropy of Corpus, $H(V_D)$, as a measure of term diversity across a collection of documents (corpus). Unlike prior works that focus on document-level entropy [8], we investigate diversity at the collection level: we will not take into account a term-frequency within a document. Indeed, this would end up penalizing terms that characterize only a portion of the corpus, moving the metric away from its target.

Definition Let $d = \{t_1, \ldots, t_n\}$ represent a document consisting of terms $t_1, \ldots, t_n$, and let $D = \{d_1, \ldots, d_N\}$ be a corpus. The vocabulary V_D represents the collection of unique terms in the corpus.

The probability $p_D(t)$ of encountering a term t while reading a document $d \in D$ is defined as:

$$p_D(t) := \frac{|\{d \in D : t \in d\}|}{|D|} \tag{1}$$

A term can be either present or absent in a document: t follows a Bernoulli distribution with $p = p_D(t = 1)$. Shannon entropy [9] can be used to express the expected surprisal of term t as:

$$H(t) = p_t \log_2 \left(\frac{1}{p_t}\right) + (1 - p_t) \log_2 \left(\frac{1}{1 - p_t}\right) \tag{2}$$

Finally, we define *corpus lexical entropy* as:

$$H(V_D) := \sum_{t \in V_D} H(t) \tag{3}$$

Notably, a term contributes zero to $H(V_D)$ if it appears in all documents, which is often the case with stop-words. Additionally, our metric remains robust to terms that never appear, as their contribution to $H(V_D)$ would be null. Furthermore, the maximum for $H(V_D)$ is $|V_D|$: a greater vocabulary leads to a higher entropy value.

Appendix A tests the metric across various corpora. Results indicate that documents in our corpus exhibit an exceptionally high degree of lexical similarity. Notably, the PubMED-Q&A [4] question subset shows 200 bits more entropy, even though questions are typically short and constrained by a structured syntactic pattern (e.g., auxiliary + subject + verb + complements).

This further highlights the severe lexical homogeneity of our dataset, which means that the differences between them are subtle at the word level. In this context, modifiers such as adjectives and negations become crucial in distinguishing repairable from non-repairable cases. Although these elements play a key role in classification, they do not introduce major shifts in the overall semantic domain (*gas pipe repairs*). This distinction poses challenges for similarity models, as some may underemphasize these key linguistic cues.

3.2. Mathematical Formulation

Let $D \subseteq \mathcal{X}^*$ be our corpus, where $\mathcal{X}$ is a finite alphabet of symbols. Let $q \in D$ be the input to the system, while $M \subseteq D$ is a set of documents that represent the memory of the system. Let $e : \mathcal{X}^* \to \mathbb{R}^n$ be the encoding function that maps a document into an n-dimensional space. Finally, consider $s : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, a similarity score between two documents.
The response of the RAG system to q is given by:

$$L \sim G_\theta \left(q, \mathcal{R}_{e,s,k,M}(q)\right) \tag{4}$$

where retrieval happens first:

$$\mathcal{R}_{e,s,k,M}(q) = \underset{\mathcal{N} \subseteq M : |\mathcal{N}| = k}{\operatorname{argmax}} \sum_{d \in \mathcal{N}} s\left(e(q), e(d)\right) \tag{5}$$

We denote the resulting set of k retrieved documents for a given query q as $\mathcal{N}^*_{q,k}$.
The generative model then conditions on $\mathcal{N}^*_{q,k}$:

$$G_\theta(q, \mathcal{N}^*_{q,k}) = \mathbb{P}_\theta(L | q, \mathcal{N}^*_{q,k}) \tag{6}$$

We make use of prompt engineering to constrain L into the ***YES/NO*** event space. Therefore, we model $L : \{\texttt{YES, NO}\} \to \{0, 1\}$ as a Bernoulli random variable:

$$L \sim \mathcal{B}(p), \quad p = \mathbb{P}_\theta(L = 1 | q, \mathcal{N}^*_{q,k}) \tag{7}$$

where p is the probability of q being patchable ('YES').

The focus of this research is to find, for the gas pipe repairs corpus D, the best combination for e, s, M (retriever), k (number of examples), then θ (generator). We recall that our entire procedure is training-free: we search θ and e among foundation models.

3.3. Retriever

The retrieval system $\mathcal{R}$ lies on three pillars: embedding function e, similarity score s, memory M.
We detail encoder selection and memory downsampling in the further sections. For the similarity function, we select the internal dot product $< \cdot, \cdot >$ as in [5].

3.3.1 Encoder e

The high level of semantic similarity caused by the lexical resemblance poses a challenge for encoders. The dataset consists of descriptions derived from tabular data. A preliminary data analysis (Table 1) highlights that the *patchability* of leaks is predominantly influenced by adjectives (e.g. *low* vs. *high* pressure) and negations (e.g. *[no]* corrosion, *[no]* walls), whereas other features such as the, damage type, play a less significant role.

Referring to the examples in our functional diagram (Figure 1), we note that, from a logical standpoint, d_1 and d_2 do not contradict q, while d_k does, as it modifies the pressure level. However, from a semantic perspective, d_1 and d_2 are more distant from q due to the presence of terms such as *Aerial*, which alters the spatial semantics, and *Polyethylene*, which modifies the material semantics.

This observation suggests that encoders pre-trained on Natural Language Inference (NLI) tasks may outperform State-of-the-Art Semantic Search models, which commonly serve as the foundation for modern retrieval systems. To test this hypothesis, we select SBERT-NLI, the most fundamental NLI-based sentence embedder, hereafter referred to as SBERT. We compare it against the top-performing models for both Semantic Search and Sentence Embeddings, as listed on the SBERT.net leaderboard: `all-mpnet-base-v2` (denoted as MPNET) and `multi-qa-mpnet-base-dot-v1` (denoted as QA-MPNET). Given that our task involves symmetric search, whereas QA-MPNET is optimized for asymmetric search—where the query and document are not interchangeable—we will expect that QA-MPNET will yield inferior performance.

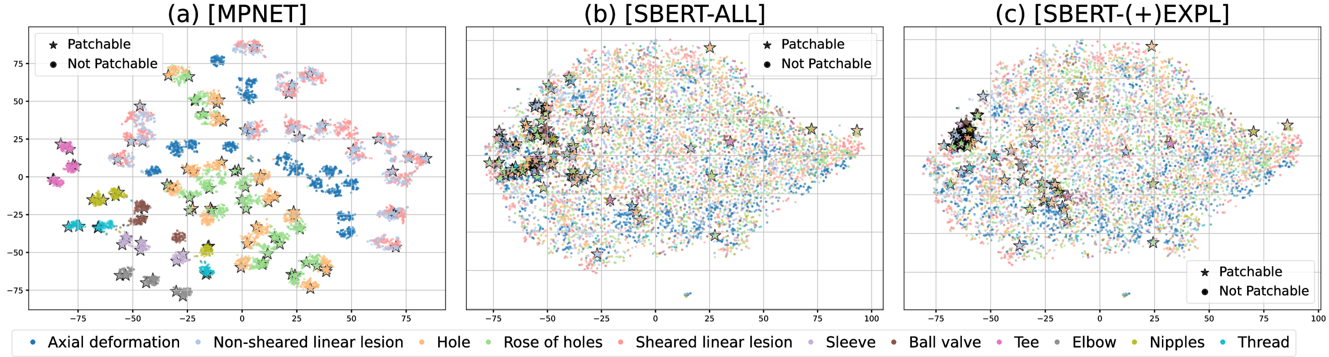

Figure 3. **t-SNE visualization of embedding spaces**. Colors indicate values for *damage type*, while stars represent patchable points, enlarged for visualization purposes. (a) MPNET effectively captures small semantic differences, distinguishing `axial deformations` from `linear lesions`. However, it neglects the impact of grammatical modifiers, failing to tell `sheared` from `non-sheared` linear lesions. (b) SBERT, on the other hand, accumulates patchable points together, offering a better representation of repairable cases. c) t-SNE preserves local structures; after removing [−]es from the [+]es neighborhoods, the remaining [+]es cluster more tightly.

As encoding function e baseline, we employ a sparse-vector representation based on Bag of Words (BoW) which accounts only for presence or absence of a term t in a document d.

3.3.2 Memory M

In real-world applications, the memory M is typically used in its entirety; however, our synthetically generated dataset may reflect an initial emphasis on optimizing Madflex® technology, resulting in a heavily skewed distribution in favor of bad outcomes ([-]es). We hypothesize that reducing the density of negative samples around positive ones might yield a more balanced context for the decoder, which ultimately makes the final decision using its domain knowledge. To this end, we propose *positive explosions* $[+]EXPL$, an algorithm that removes negative samples within a specified radius around positive points. The name of the algorithm originates from an analogy with an explosion of radius r, that wipes out anything within that range.

For each positive sample $p \in M_+$, we first retrieve its k-nearest neighbors $\mathcal{N}^*_{p,k}$ and then select only the unsuccessful repairs $\mathcal{N}^{*,-}_{p,k}$. Using the L2 norm for distance calculations, we define

$$r_k = \frac{1}{|M_+|}\sum_{p\in M_+}\sum_{n\in\mathcal{N}^{*,-}_{p,k}} \|e_p - e_n\|_2 \tag{8}$$

We choose the final explosion radius $r_{\texttt{EXPL}}$ at the point where the rate of change in r_k is maximal,

$$r_{\texttt{EXPL}} := \max_{k\in K\setminus\{1\}} \left(\frac{r_{k-1}+r_k}{2}\right) \tag{9}$$

with $K = \{1, \ldots, 11\}$.

We test both full-memory and downsampled approach, respectively named `ALL` and `[+]EXPL`.

3.4. Decoder θ

The decoder is responsible for labeling the query q by means of q itself, the retrieved documents $\mathcal{N}^*_{q,k}$, and the domain knowledge embedded in θ. The selection of the best decoder, in our work, entails choosing the most suitable model from a set of open-source foundation models based purely on their intrinsic domain knowledge.

We evaluate three models: Llama3.2, Llama2-13B, and Mistral7B. The evaluation is done in a zero-shot setting to isolate the model- *domain knowledge*. This approach allows us to determine which model has the best understanding about gas leaks patchability.

Once the best-performing model is selected, we proceed to a classical few-shot learning phase. During this phase, a random subset $E \subset M$ of the retrieval memory is provided to the model to give it some contextual guidance. This step establishes a baseline for the impact of the retrieval system: if the performance of the retrieval system falls below this baseline, it will be deemed ineffective.

4. Experiments

Experiments were conducted on a Colab notebook with a T4 GPU, using Ollama to access autoregressive models. Due to GPU disconnections on Colab free plan, we used the full query set Q for retriever experiments but had to downsample negative cases when running predictions with an autoregressive model. This resulted in 100 non-repairable and 25 repairable cases. Model performance is evaluated using the F1-MACRO score to ensure balanced assessment across both classes.

4.1. Data

The dataset comprises both boolean and categorical features, which are used to generate textual descriptions encapsulating the conditions of each pipe case at the time of damage. These descriptions provide no information about the repair process. The repair outcome, indicating suc-

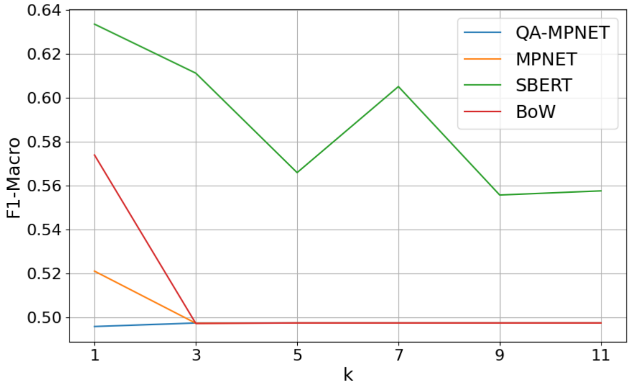

Figure 4. **k-NN Classification** for three embedding models (SBERT, MPNET, QA-MPNET) and a sparse vector representation (BoW). SBERT consistently outperforms all other models, with BoW surpassing both MPNETs at $k = 1$. After $k = 1$, all models except SBERT drop to 0.50 F1-MACRO, suggesting that positive labels are not represented as the majority in a group of k elements. The superior performance of BoW over MPNETs highlights the impact of grammatical modifiers on documents labeling.

cess or failure, is recorded in the `Successful` field as `True` or `False`. The corpus consists of 11 904 examples, each described by 15 features. Among them, 98.47% (11 778 cases) correspond to unsuccessful patch attempts, while only 1.06% (126 cases) are successful repairs. To ensure proportional representation of both positive and negative cases, we applied stratified sampling to split the dataset into two subsets: 80% for the RAG system memory (M_{ALL}) and 20% for the query set Q, which serves the purpose of being the test set. M_{ALL} consists of 101 positive cases and 9422 negative cases, while the query set contains 25 positive cases and 2356 negative cases. Whenever an autoregressive model is employed to make a classification, Q is downsampled to Q_d by shrinking from 2356 down to 100, the number of non-patchable cases due to resource limitations.

Table 1 presents a subset of the binary features that define the dataset. For each feature, the table indicates the values typically associated with patchable and non-patchable cases. This helps illustrate how different conditions influence outcomes.

Table 1. Feature Values by Patchability Class

Feature	**Patchable [+]**	**Not Patchable [-]**
Bad Condition	Yes/No	Yes/No
Severe Corrosion	No	Yes/No
Pipe Covered	No	Yes/No
Faulty Branch	No	Yes/No
High Pressure	No	Yes/No
Damaged Valve	No	Yes/No
Ribs	No	Yes/No

4.2. Encoder e

Figure 4 presents encoder selection results, evaluated using the F1-MACRO score over Q by means of a k-Nearest-Neighbors (k-NN) classification, where similarity score was $s =< \cdot, \cdot >$. **SBERT** achieves the highest **F1-MACRO** score of **0.63** and is the only encoder capable of correctly classifying repairable cases beyond $k = 1$.

This aligns with our hypothesis: reparability is primarily determined by logical operators (such as negations and degree modifiers) rather than material properties or spatial features. These logical operators critically influence gas leaks patchability, as seen in the need for a *low* pressure level and *no* ribs. We attribute weaker performance of the MPNet to the *permuted* nature of a part of its training objective: a model trained to reconstruct meaning from shuffled fragments may deprioritize negation and degree modifiers, treating them as non-essential from a semantic perspective. ***SBERT** is selected as the encoder for our system, given its optimal performance and alignment with our task..*

4.3. Decoder θ

We evaluate gas repair knowledge of three different LLMs: Llama3.2, Llama2-13B, and Mistral7B. The evaluation focuses on **F1-MACRO** and **Self-Consistency (SC)** scores.

Self-Consistency (SC) measures the stability predictions of a model by evaluating whether it provides identical answers across multiple runs. Given a set of queries T, we define SC as:

$$SC = \frac{\sum_{q \in T} I(G_\theta(q)^{(1)} = G_\theta(q)^{(2)} = \cdots = G_\theta(q)^{(n)})}{|T|} \tag{10}$$

where $G_\theta(q)^{(i)}$ represents the model prediction for query q in the i-th run, and the indicator function $I(\cdot)$ returns 1 if all predictions are identical across n runs, and 0 otherwise. As pointed out previously, in this work $T = Q_d$. Results are shown in Table 2, while the prompt is reported in Appendix B.

Table 2. Model Performance Comparison

Model	**Llama3.2**	**Llama2-13B**	**Mistral 7B**
Avg(F1)	0.62	0.44	**0.68**
Self Consistency	0.75	1.00	**0.86**

Mistral 7B demonstrates the best balance of accuracy and consistency, making it the most reliable decoder model for our repair classification task. It achieves an **Avg(F1)** of **0.68** and a **Self-Consistency (SC)** score of **0.86**, indicating that it not only provides accurate predictions but also maintains stability across multiple runs. Notice that the decoder in zero-shot achieves better performance than the retrieval system alone.

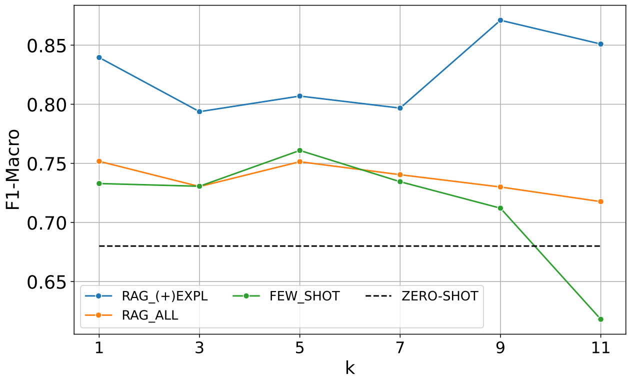

Figure 5. **RAG Evaluation Summary**. Retrieval boosts performance when backed by $M = [+]EXPL$, highlighting the importance of data curation in unbalanced datasets where terminology overlap is high.

4.4. Retrieval-Augmented Generation

We establish a baseline for **Mistral 7B** in few-shot learning by sampling E_k random points from M_{ALL}, maintaining a ratio, where for a given k, the number of non-patchable (negative) cases is $\lfloor \frac{2}{3}k \rfloor$ and the number of patchable (positive) cases is $\lceil \frac{1}{3}k \rceil$. Examples $E_k \in M_{\text{ALL}}$ are fixed throughout the experimentation. If retrieval-augmented predictions fail to exceed this baseline, then retrieval provides no additional benefit. Using the same prompt for RAG with passing random examples, we test $k \in \{1, 3, 5, 7, 9, 11\}$ and find the best performance at $k = 5$ (0.76 F1-macro).

For our RAG experiments, we compare M_{ALL} and $M_{[+]EXPL}$, using e= SBERT-NLI and θ= Mistral 7B, referring to these configurations as *RAG-ALL* and *RAG-[+]EXPL*. The prompt used for retrieval is detailed in the Appendix B. Performance is evaluated across $k \in \{1, 3, 5, 7, 9, 11\}$, with results depicted in Figure 5.

With a top F1-Marco score of **0.87** reached at **k = 9**, **RAG-[+]EXPL** is the best solution presented in this work. Notably, it outperforms all other settings regardless of the number of examples retrieved (k), while *RAG-ALL* fails to consistently surpass the few-shot baseline. This highlights that accurate memory selection M in low-lexical-entropy corpus D is essential, to provide the right examples to the decoder: indiscriminate memory inclusion may introduce noise rather than useful context.

To quantify improvements, we compare zero-shot, few-shot, and the best-performing RAG configuration. Figure 6 shows that few-shot surpasses zero-shot by 10.50%, while retrieval-based augmentation further boosts performance by 14.47%. This demonstrates that Mistral 7B benefits from additional context, even when not directly entailed to q, with the most significant gains achieved through accurate retrieval. *The best configuration explored in this work is:*

- *$\mathcal{R}$: e=SBERT-NLI, $s =< \cdot, \cdot >$, $M = M_{[+]EXPL}$;*
- *$k = 9$;*
- *θ =MISTRAL7B.*

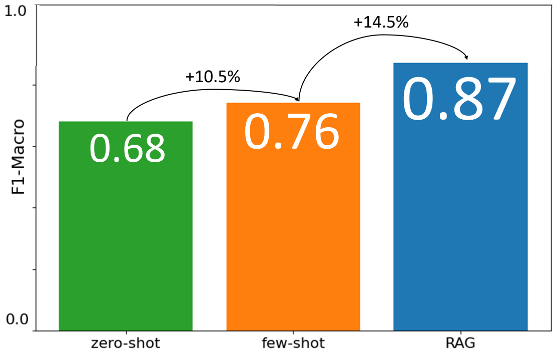

Figure 6. **RAG improves performance**. Few-shot achieves 0.76 F1 at $k = 5$, while RAG reaches its best result at $k = 9$ with $M = [+]EXPL$, setting the highest score in our study. Mistral 7B is used as the decoder θ in all configurations.

5. Conclusion

This paper examined the application of Retrieval-Augmented Generation (RAG) in low-lexical-variety settings to classify gas pipe damage descriptions based on their patchability. We introduced a metric to evaluate the level of lexical diversity within a corpus. Our methodology combined SBERT for retrieval with Mistral 7B for classification generation, effectively addressing the challenges associated with high semantic similarity and class imbalance. We proposed an algorithm for downsampling to clean up neighbors when they are too crowded ([+]EXPL), improving the probability of extracting examples from the class with fewer samples, increasing the likelihood of these examples being passed to the decoder. The results show that the RAG system benefits from this technique (F1-Macro = 0.87), as the decoder sees a more diversified set of examples. This work highlights the importance of NLI-trained encoders in semantically constrained datasets and the effectiveness of targeted downsampling in mitigating class imbalance and bias. Future directions include combining SBERT and MPNet to develop advanced RAG systems, enhancing retrieval and classification performance. Additionally, we plan to evaluate the system using real-world data in multiple languages, such as Italian, and investigate the rationale that the model provides in addition to the classification.

6. Project Source Code

The project codes can be found in the following public GitHub repository: ***2024-P10-RAG-GAS***.

References

[1] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*, 2023.

[2] Italgas. Patch madflex, n.d. Retrieved November 1, 2024.

[3] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[4] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering, 2019.

[5] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[6] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[7] Alex Reinhart, David West Brown, Ben Markey, Michael Laudenbach, Kachatad Pantusen, Ronald Yurko, and Gordon Weinberg. Do llms write like humans? variation in grammatical and rhetorical styles. *arXiv preprint arXiv:2410.16107*, 2024.

[8] Pablo Rosillo-Rodes, Maxi San Miguel, and David Sanchez. Entropy and type-token ratio in gigaword corpora. *arXiv preprint arXiv:2411.10227*, 2024.

[9] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[10] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867, 2020.

A. Corpus Lexical Entropy

Table 3 shows the effectiveness of our metric. As expected, the Common Crawl-News (CC-N) corpus has the highest entropy, reflecting the inherent diversity of news articles. Domain-specific corpora such as PubMED-Q&A [4] have lower $H(V_D)$ values since all ds deal with a specific topic, i.e. medicine. Questions (PM-Q) tend to be shorter than answers (PM-A). Also, questions follow a precise syntactical structure that makes them less prone to variety, as $H(V_D)$ correctly reflects. $H(V_{\tilde{D}})$ evaluates the lexical entropy of a corpus by downsampling the original dataset D to match the size of our own collection, denoted as $\tilde{D}$. The idea is to see how well the metric still reflects the lexical diversity of the documents even when the corpus is downsampled. From T3 one can see that despite reducing the size of the corpus, the metric remains sensitive to the differences between corpora, still reflecting their intrinsic level of lexical diversity.

Table 3. Corpora Lexical Entropies (10^3)

Stats	CC-N	PM-A	PM-Q	Ours
$\lvert D\rvert$	700	60	60	11
$\lvert V\rvert$	884	50	31	2
$H(V_D)$	5.23	1.08	0.55	0.32
$\lvert V_{\tilde{D}}\rvert$	102	15	24	2
$H(V_{\tilde{D}})$	4.83	1.03	0.52	0.32

B. Prompt Templates

Zero-Shot
You are a pipeline maintenance assistant with expertise in gas pipe repairs.
Your task is to determine if a pipe leak can be repaired by patches.
Answer selecting one of the following options:
- ”YES” if the damage can be repaired by patches;
- ”NO” if the damage cannot be repaired by patches.

do not write any word but either YES or NO

Damage Description: {description}

Answer:

RAG (and few-shot)
You are an expert in gas pipeline damage analysis. Your goal is to classify whether a given damage description is ”patchable” (YES) or ”not patchable” (NO).

Here is the damage description you need to classify:
- Query: {description}”

Here are some examples:
{examples}

Task:
1. Analyze the similarities and differences between the query and the retrieved cases.
2. Based on your analysis, answer by selecting **one and only one** among the following two labels:
- **YES** if the damage can be repaired by patches;
- **NO** if the damage cannot be repaired by patches.

YOUR ANSWER MUST INCLUDE ONLY AND ONLY THE LABEL: do not write your analysis or any other word besides the label **YES** or the label **NO.**

Label: