Aalto-yliopisto
Aalto-universitetet
Aalto University

Master's Programme in ICT innovation (EIT Digital Master School)

Semantic Matching Under Constraints

Retrieval vs. Classification Approaches for Low-Resource, Structurally Homogeneous, Fixed-Category Scenarios

Ying Xu

Master's thesis
2025

Aalto-yliopisto
Aalto-universitetet
Aalto University

Author Ying Xu

Title of thesis Semantic Matching Under Constraints: Retrieval vs. Classification Approaches for Low-Resource, Structurally Homogeneous, Fixed-Category Scenarios

Programme ICT innovation (EIT Digital Master School)

Major Data Science

Thesis supervisor Marrit Korpi-Lagg

Thesis advisor(s) Dr Amirhossein Layegh Kheirabadi and Professor Henrik Boström

Collaborative partner Munters AB

Date 07.23.2025 **Number of pages** 57 **Language** English

Abstract

This study investigates the comparative effectiveness of retrieval-based and classification based approaches to semantic matching task in low-resource, fixed-category, and structurally homogeneous text scenarios, using industrial fault diagnosis scenario as an example. The core task is to match free-form user fault descriptions to standardized fault rule texts.

Five experimental configurations are evaluated in this study, combining dense retrieval, semantic reranking, and domain-specific Named Entity Recognition (NER) as comparison strategies. The analysis is structured around four key dimensions: 1) performance trade-offs between retrieval and classification methods; 2) the effectiveness of semantic reranking in refining embedding-based retrieval results; 3) the added value of NER-based preprocessing for semantic alignment, including comparisons of tagging strategies and application scopes; and 4) the impact of varying user input types (lengthy, precise, and vague) on model robustness.

Results demonstrate that classification-based methods achieve superior performance with significantly higher accuracy compared to retrieval approaches, due to their ability to fully leverage labeled training data in constrained scenarios. Semantic reranking substantially improves retrieval accuracy but introduces considerable computational overhead. Domain-specific NER preprocessing shows limited effectiveness and may degrade performance due to domain adaptation mismatch and information dilution. Notably, negative interactions between system components can occur, where combining enhancements yields inferior results compared to individual components. The analysis also reveals that lengthy, conversational user descriptions outperform precise technical specifications, explained through semantic over-specification and pathway redundancy frameworks.

These findings challenge conventional assumptions about component integration in semantic matching systems and provide insights for designing domain-specific applications where labeled data is limited and user input quality varies.

Keywords Semantic Matching, Classification-based Method, Retrieval-based Method, NER, Low resource

Degree Project in Computer Science and Engineering
Second cycle, 30 credits

Semantic Matching Under Constraints

Retrieval vs. Classification Approaches for Low-Resource, Structurally Homogeneous, Fixed-Category Scenarios

YING XU

Semantic Matching Under Constraints

Retrieval vs. Classification Approaches for Low-Resource, Structurally Homogeneous, Fixed-Category Scenarios

Ying Xu

Master's Programme in ICT Innovation (120 credits)
Supervisor: Amirhossein Layegh Kheirabadi
Examiner: Henrik Boström
School of Electrical Engineering and Computer Science
Host company: Munters AB
Swedish title: Semantisk matchning under begränsningar: Hämtning kontra klassificeringsmetoder för scenarier med låga resurser och fast kategori

Abstract

This study investigates the comparative effectiveness of retrieval-based and classification-based approaches to semantic matching task in low-resource, fixed-category, and structurally homogeneous text scenarios, using industrial fault diagnosis scenario as an example. The core task is to match free-form user fault descriptions to standardized fault rule texts.

Five experimental configurations are evaluated in this study, combining dense retrieval, semantic reranking, and domain-specific Named Entity Recognition (NER) as comparison strategies. The analysis is structured around four key dimensions: 1) performance trade-offs between retrieval and classification methods; 2) the effectiveness of semantic reranking in refining embedding-based retrieval results; 3) the added value of NER-based preprocessing for semantic alignment, including comparisons of tagging strategies and application scopes; and 4) the impact of varying user input types (lengthy, precise, and vague) on model robustness.

Results demonstrate that classification-based methods achieve superior performance with significantly higher accuracy compared to retrieval approaches, due to their ability to fully leverage labeled training data in constrained scenarios. Semantic reranking substantially improves retrieval accuracy but introduces considerable computational overhead. Domain-specific NER preprocessing shows limited effectiveness and may degrade performance due to domain adaptation mismatch and information dilution. Notably, negative interactions between system components can occur, where combining enhancements yields inferior results compared to individual components. The analysis also reveals that lengthy, conversational user descriptions outperform precise technical specifications, explained through semantic over-specification and pathway redundancy frameworks.

These findings challenge conventional assumptions about component integration in semantic matching systems and provide insights for designing domain-specific applications where labeled data is limited and user input quality varies.

Keywords

Semantic Matching, Classification-based Method, Retrieval-based Method, NER, Low-resource

Sammanfattning

Denna studie undersöker den jämförande effektiviteten hos hämtningsbaserade och klassificeringsbaserade metoder för semantiska matchningsuppgifter i scenarier med låga resurser, fast kategori och strukturellt homogena texter, med hjälp av industriella feldiagnosscenarier som exempel. Kärnuppgiften är att matcha fritt formulerade användarfelbeskrivningar med standardiserade felregeltexter.

Fem experimentella konfigurationer utvärderas i denna studie, där tät hämtning, semantisk omrankning och domänspecifika NER kombineras som jämförelsestrategier. Analysen är strukturerad kring fyra nyckeldimensioner: 1) prestandaavvägningar mellan hämtnings- och klassificeringsmetoder; 2) effektiviteten av semantisk omrankning för att förfina inbäddningsbaserade hämtningsresultat; 3) mervärdet av NER-baserad förbehandling för semantisk justering, inklusive jämförelser av taggningsstrategier och applikationsomfång; och 4) effekten av varierande användarinmatningstyper (långsamma, precisa och vaga) på modellens robusthet.

Resultaten visar att klassificeringsbaserade metoder uppnår överlägsen prestanda med betydligt högre noggrannhet jämfört med hämtningsmetoder, på grund av deras förmåga att fullt ut utnyttja märkta träningsdata i begränsade scenarier. Semantisk omrankning förbättrar hämtningsnoggrannheten avsevärt men introducerar avsevärd beräkningsoverhead. Domänspecifik NER-förbehandling visar begränsad effektivitet och kan försämra prestandan på grund av domänanpassningsmissmatchning och informationsutspädning. Det är värt att notera att negativa interaktioner mellan systemkomponenter kan uppstå, där kombination av förbättringar ger sämre resultat jämfört med enskilda komponenter. Analysen visar också att långa, konversationsbaserade användarbeskrivningar överträffar exakta tekniska specifikationer, vilket förklaras av semantisk överspecifikation och redundansramverk för vägar.

Dessa resultat utmanar konventionella antaganden om komponentintegration i semantiska matchningssystem och ger insikter för att utforma domänspecifika applikationer där märkt data är begränsad och användarinmatningskvaliteten varierar.

Nyckelord

Semantisk matchning, Klassificeringsbaserad metod, Hämtningsbaserad metod, Namngiven entitetsigenkänning, Lågresursanvändning

Acknowledgements

I am sincerely grateful to my examiner, Henrik Boström, for your patient guidance and thoughtful feedback. You helped me stay on the right track and gave me the confidence to keep moving forward. I also want to thank my university supervisor, Amirhossein Layegh Kheirabadi, for your continuous support throughout this study. From choosing the topic to designing the study and writing the thesis, your advice and encouragement have helped me grow on my academic journey.

Thanks to the EIT Digital Master School, Aalto university and KTH for offering me a valuable platform for learning and exchange. This experience not only broadened my view of the world but also provided important resources and support for this research. Also, I would like to thank my company, Munters, for giving me the chance to learn and grow in a real-world environment. I'm especially thankful to my mentors at the company for generously sharing their time, knowledge, and advice. I also appreciate the support and collaboration from my colleagues and fellow interns.

I am deeply thankful to my family, especially my parents, for always standing behind me. Your quiet support and encouragement gave me peace of mind and the strength to keep going. To my friends—thank you for being there during the hard times, for comforting me when I was anxious, and for celebrating the small wins with me. You brought warmth and meaning to my life during this research journey. I also want to thank myself, for not giving up, for taking every step seriously, and for learning and growing no matter the circumstances. I hope to keep learning and becoming a better version of myself.

Thanks to the researchers and scholars whose work laid the foundation for this study. I deeply respect your contributions to knowledge and progress.

Finally, I want to thank all the people who have always been there for me. Your support, care, and presence have meant a lot to me.

Contents

Acronyms

CNNs Convolutional Neural Networks. 10

GNNs Graph Neural Networks. 2

KGs Knowledge Graphs. 9

LLM Large Language Models. 20, 21

LSTMs Long Short-Term Memory. 10

ML Machine Learning. 9

NER Named Entity Recognition. , 3–6, 9, 17–22, 24–32, 34, 36–39, 44, 45

NLP Natural Language Processing. 1, 7, 15, 38

PLMs Pre-trained Language Models. 10

QA Question Answering. 1, 7, 20

RAG Retrieval-Augmented Generation. 1, 7

RNNs Recurrent Neural Networks. 10

SVM Support Vector Machines. 9

TF-IDF Term Frequency-Inverse Document Frequency. 7, 9

Tree-RNNs Tree-structured recursive neural networks. 2

1 Introduction

1.1 Background

Semantic matching represents a fundamental challenge in Natural Language Processing (NLP) that focuses on determining the semantic similarity or relatedness between two pieces of text. This capability enables computational systems to bridge surface-level lexical differences and identify shared meaning, even when the wording varies significantly [17]. The importance of semantic matching extends across numerous applications in modern information systems, including Question Answering (QA), dialogue systems, information retrieval, and recommendation engines. In knowledge base QA, where natural language questions are mapped to structured repositories of factual information (i.e., knowledge bases), semantic matching plays a pivotal role. Semantic matching is crucial here for aligning user queries—often expressed in free-form language—with these structured representations, enabling accurate question interpretation and answer retrieval [2]. Similarly, in information retrieval applications, semantic matching addresses the fundamental term mismatch problem by enabling the retrieval of conceptually similar content even when exact keyword matches are absent [17, 8].

The practical significance of semantic matching becomes particularly evident in specialized domains such as industrial fault diagnosis with the proliferation of Industry 4.0 applications, where users may describe machine-related issues using vastly different terminology. For instance, expressions like ”unusual machine noise” and ”strange sound during operation” refer to the same underlying problem but require sophisticated semantic understanding to be correctly matched to standardized fault categories. This challenge exemplifies the need for robust semantic matching systems that can handle domain-specific language variations while maintaining high precision in tasks.

Contemporary approaches to semantic matching generally fall into two primary paradigms: retrieval-based methods and classification-based methods [2]. Retrieval-based approaches typically employ dense vector representations and similarity computation techniques, often implemented within Retrieval-Augmented Generation (RAG) frameworks [14]. These methods excel in handling dynamic datasets and open-domain scenarios where flexibility and scalability are especially important. Conversely, classification-based methods treat semantic matching as a supervised learning problem where predefined categories serve as class labels, enabling precise boundary learning through labeled training data [7].

1.2 Problem

Despite substantial advances in semantic matching research, current approaches encounter critical limitations when deployed in real-world constrained environments. Several critical research gaps persist in the current literature:

Limited Comparative Analysis Under Realistic Constraints. Despite strong results for both retrieval-based and classification-based paradigms, comparative studies are typically limited to narrowly defined domains [28]. This is largely because semantic matching tasks are highly context-dependent, different domains often require different modeling assumptions and representations, making general-purpose comparisons less meaningful. As a result, most research evaluates the two paradigms in isolation within specialized application settings, and systematic cross-paradigm evaluations under generalizable or practical constraints remain scarce [28]. A study by Cohen et al. [6] examined classification and retrieval methods for document triage tasks in the biomedical domain, finding that classification approaches can outperform retrieval methods when supported by careful feature engineering and sufficient labeled data. Similarly, research by Wale et al. [29] investigated how different descriptor spaces influence both retrieval and classification performance in chemical compound analysis, demonstrating that representation choices significantly affect both paradigms. However, their investigation was constrained to structured and extensively annotated datasets.

Secondly, **insufficient attention to low-resource scenarios:** Most comparative studies concentrate on resource-abundant environments [12]. For instance, research on semantic matching in e-commerce search environments involves product search applications with 50,000 queries each [26], providing abundant data resources for detailed comparative analysis of trade-offs between accuracy and efficiency when selecting classification versus retrieval models. While some recent work addresses low-resource challenges through data augmentation techniques, such as the PDAM-FAQ [30] framework that focuses on FAQ semantic matching problems in low-resource scenarios and proposes an augmentation framework to improve performance through two key modules: data augmentation and feature fusion semantic matching models, achieving promising results, these studies focus on improving individual methods rather than conducting systematic cross-paradigm comparisons.

Then, **Structural Homogeneity Challenge:** Many practical applications involve structurally homogeneous text characterized by highly similar grammatical structures, repetitive sentence patterns, and consistent terminology usage on one side of the matching task [18]. In industrial troubleshooting scenarios, fault descriptions typically follow similar linguistic patterns and structural templates, resulting in semantically similar vector representations that challenge differentiation. For example, user descriptions such as "machine makes strange noise during startup," "equipment produces unusual sound when starting," and "abnormal noise occurs at machine initialization" share nearly identical grammatical structures and exhibit minimal lexical variation, leading to highly similar embeddings despite potentially referring to different underlying fault mechanisms. Simi- larly, descriptions like "system shows error code," "display indicates fault message," and "screen presents warning signal" follow parallel structural patterns that compress into similar vector spaces, making fine-grained semantic distinction difficult. This structural homogeneity creates dense clusters in the embedding space where semantically distinct fault descriptions become difficult to separate, challenging both retrieval-based similarity computation and classification-based boundary learning [18]. Existing domain-specific re- search acknowledges this challenge but approaches it from single-paradigm perspectives. One effective strategy involves explicit encoding of syntactic structures by incorporating syntactic trees or dependency parses into text representations instead of relying solely on simple weighted averages of word embeddings [37]. Techniques such as Graph Neural Networks (GNNs) can be employed to encode syntactic trees by utilizing dependency trees to establish connections between syntactically related words and directly exploit long-range relations [10]. Tree-structured recursive neural networks (Tree-RNNs) can also be employed to learn sentence representation by exploiting syntactic structures [34], enabling the model to capture subtle structural differences that are crucial for distinguishing semantically different but structurally similar sentences.

The convergence of these limitations reveals a critical gap: the absence of systematic comparative analysis between classification-based and retrieval-based semantic matching approaches under realistic deployment conditions characterized by limited resources, fixed categories, and structurally homogeneous text. This leads to the central research question of this study: **In low-resource, fixed-category, and structurally homogeneous text scenarios, which approach yields superior performance in semantic matching: classification-based methods or retrieval-based methods?**

This gap is particularly problematic for industrial applications where practitioners must make architectural decisions without adequate empirical evidence. The choice between paradigms significantly impacts both development costs and operational performance, yet existing research provides insufficient guidance for these decisions.

This study is distinguished by:

1. First, it presents the first systematic comparison between retrieval-based and classification-based approaches under realistic constraints, including low-resource availability, structural homogeneity, and fixed-category domains.
2. Second, it identifies and explains the conditional effectiveness of NER-based enhancement, challenging the assumption of its universal benefit.
3. Third, it reveals a counterintuitive pattern whereby precise user inputs may underperform compared to vague descriptions, due to overfitting and semantic mismatch.
4. Fourth, it uncovers negative interaction effects when combining multiple enhancement components, such as NER and reranking.
5. Finally, the study proposes a deployment-oriented decision framework to support architectural choices in constrained industrial environments.

1.3 Purpose

The purpose of this study is to address the identified knowledge gap by answering the research question: In low-resource, fixed-category, and structurally homogeneous text scenarios, which approach yields better performance in semantic matching: classification-based methods or retrieval-based methods?

This study represents the first systematic comparative analysis of these paradigms under the specified constraint conditions, addressing a critical gap between theoretical semantic matching research and practical deployment requirements in industrial environments. The research question addresses a critical decision point in system architecture where the choice between paradigms can significantly impact both development costs and operational performance.

The question encompasses two primary evaluation dimensions that reflect the core concerns of practitioners in resource-constrained environments:

- **Resource Efficiency**: How effectively each approach utilizes available computational resources and training data
- **Semantic Matching Accuracy**: The precision and effectiveness of semantic matching under specified constraints, including classification accuracy and retrieval precision

The significance of this research lies in its potential to generate actionable architectural guidance for domain-specific applications such as industrial diagnostics, healthcare decision support, and specialized customer service systems. By providing empirical evidence about the comparative performance of these approaches under realistic constraints, this study aims to inform more effective system design decisions and resource allocation strategies.

1.4 Objectives

To comprehensively address the research question, this study establishes the following specific objectives:

1.4.1 Primary Objective

To empirically evaluate and compare the performance of classification-based and retrieval-based semantic matching methods in scenarios characterized by low-resource availability (50-500 labeled examples per category), fixed-category systems, and structurally homogeneous text.

1.4.2 Secondary Objectives

- Develop representative experimental frameworks for both classification-based and retrieval-based approaches that accurately reflect real-world constraints, including limited labeled data availability and domain-specific text characteristics.
- Assess resource utilization efficiency by measuring computational requirements and data efficiency across both paradigms to provide practical guidance for resource allocation decisions.
- Evaluate semantic matching accuracy through comprehensive performance metrics including precision, recall, F1-score, and task-specific accuracy measures under varying data availability conditions.
- Analyze robustness and generalizability by examining how each approach performs across different levels of structural text variations within the constrained environment.
- Identify optimal implementation strategies for each paradigm, including preprocessing techniques (e.g., NER), model architectures, and hyperparameter configurations that maximize performance under the specified constraints.
- Generate practical recommendations for practitioners regarding method selection criteria, implementation considerations, and expected trade-offs between accuracy and resource efficiency.

These objectives are designed to provide both theoretical insights into the comparative strengths of different semantic matching paradigms and practical guidance for real-world system implementation in resource-constrained environments.

1.5 Research Methodology

This study employs a multi-factorial experimental design to systematically evaluate the comparative performance of classification-based and retrieval-based semantic matching approaches under resource-constrained conditions. The methodology incorporates three primary experimental factors: paradigmatic approach (retrieval-based versus classification-based methods), semantic enhancement strategies (including baseline configurations, NER preprocessing, reranking mechanisms, and combined approaches), and input characteristic variations (encompassing different linguistic styles and descriptive precision levels).

The research framework integrates ablation study methodologies to decompose complex system performance into constituent component contributions. This approach systematically evaluates individual components by comparing performance with and without each element while maintaining experimental control over other factors.

Five distinct experimental configurations are implemented to assess semantic enhancement strategy effectiveness: baseline embedding similarity, reranking-enhanced retrieval, NER-based retrieval, combined enhancement (NER & reranking) approaches, and direct classification methods. These configurations are designed to isolate and evaluate the specific impact of preprocessing mechanisms, reranking techniques, and classification models on overall fault identification system performance. Comprehensive experimental procedures, detailed system architectures, and implementation specifications are presented in Chapter 3.

1.6 Delimitations

This study establishes several important delimitations that define the deliberate scope and boundaries of the research investigation:

- **Language Scope:** The experimental evaluation uses English-language datasets to maintain consistency and enable controlled comparison. Multilingual considerations are beyond the current research scope.
- **Methodological Focus:** The comparative analysis is limited to established classification-based and retrieval-based semantic matching paradigms, excluding hybrid approaches to enable clear paradigm comparison. This includes the deliberate exclusion of k-Nearest Neighbors (kNN) methods with pretrained embeddings, which represent a hybrid approach combining retrieval-like flexibility (no retraining required, open label sets) with classification-like categorical outputs. While such methods offer compelling advantages, their inclusion would complicate the direct comparison between the two primary paradigms that this study aims to investigate.

1.7 Structure of the Thesis

This thesis is organized into five chapters, each of which addresses a key stage in the research process.

Chapter 1 provides an overview of the study, including the background, the specific problem being addressed, the purpose and objectives of the research, the chosen methodology, and the scope and limitations of the work. This chapter establishes the motivation and contextual foundation for the study.

Chapter 2 offers an extended background by reviewing relevant literature in the field of semantic matching. It covers the definition and formulation of the semantic matching task, presents a detailed discussion of retrieval-based and classification-based approaches, analyzes related work in the domain, and outlines the metrics commonly used for evaluation. This chapter situates the study within existing research and identifies theoretical and practical gaps.

Chapter 3 presents the research methodology. It explains the rationale behind the selection of the methodological approach and describes the technical framework of the proposed solution. This includes the architecture of the system, the integration of enhancement strategies such as domain-aware preprocessing, and the design of the experimental framework used to evaluate the models. It also outlines how the methodology aligns with the research objectives.

Chapter 4 reports the results and analysis. It begins with an overview of the overall model performance, followed by a breakdown of results based on specific components, such as different NER methods and varying input description types. The chapter also includes per-class performance analysis and a discussion section that interprets the findings in light of the research questions and prior literature.

Chapter 5 concludes the thesis by summarizing the main contributions, acknowledging the limitations of the study, and suggesting possible directions for future work. It reflects on both the theoretical insights gained and the practical implications of the research. Through this structure, the thesis aims to offer a coherent and comprehensive investigation into the semantic matching problem within domain-specific contexts.

2 Extended Background

2.1 Semantic Matching Task

Semantic matching refers to the task of determining whether two natural language texts are semantically related, similar in meaning, or refer to the same underlying entity or concept [17].

Formally, given a query text Q and a set of candidate texts $C = \{C_1, C_2, \ldots, C_N\}$, the goal of semantic matching is to find the candidate text C_i that is most semantically similar or related to Q. This is often framed as learning a similarity function $S(Q, C_i)$ that outputs a score indicating the degree of semantic relatedness.

Semantic matching is a foundational component in many NLP applications, including question–answer pairing in QA systems, intent identification in recommendation systems, intent matching in dialogue systems, and, in the context of this study, matching user input to predefined rule sets such as alarm or fault-handling protocols.

As mentioned before, contemporary semantic matching methods are broadly categorized into two types: retrieval-based and classification-based approaches [2].

For retrieval-based approaches, the task is formulated as ranking candidates from a large corpus based on similarity scores: given query Q and candidate set C, find $\arg\max_{C_i \in C} S(Q, C_i)$ where $S(\cdot, \cdot)$ measures semantic similarity. This enables handling open-set scenarios and scaling to large corpora.

For classification-based approaches, the task is formulated as predicting discrete semantic relationships: given query-candidate pair (Q, C_i), predict label $y = f(Q, C_i; \theta)$ indicating whether they match or their relationship type. This focuses on modeling pairwise interactions within fixed candidate sets.

The following provides a concise overview of the commonly employed techniques within each category.

2.2 Retrieval-Based Methods

Retrieval-based methods, often implemented within the RAG framework[14], typically encode text content into computable vectors and measure their semantic similarity.

2.2.1 Core Stages and Initial Challenges

Initially, the core stages of retrieval-based methods primarily involved:

- **Text Embedding Generation**: This initial phase focuses on transforming both the input query and all candidate texts into dense numerical vectors, known as semantic embeddings. These embeddings are designed to capture the deep semantic meaning of the text, ensuring that semantically similar texts are positioned close to each other in the vector space [24]. Early approaches might have relied on statistical methods like Term Frequency-Inverse Document Frequency (TF-IDF), or word embedding models such as Word2Vec [21] or GloVe [22], which generated sentence or document-level representations by averaging or weighting word vectors. With the advent of deep learning, Transformer-based pre-trained language models (e.g., BERT [9], RoBERTa [19]) became capable of producing more context-aware embeddings, significantly improving text representation quality.

- **Semantic Similarity Computation**: Once text embeddings are obtained, the system calculates a similarity score between the query embedding and each candidate text embedding to quantify their semantic relevance [14]. Common similarity metrics include cosine similarity and dot product. These measures efficiently rank large sets of candidate texts, quickly identifying potentially relevant results [14].

However, in practical applications, especially when dealing with large-scale and semantically complex text data, relying solely on these two core stages quickly revealed limitations. The primary difficulties encountered were:

- **Coarse-grained" Matching Issues**: The initial text embeddings and similarity computations often captured only the general semantic direction or topic relevance of texts. They struggled to distinguish subtle semantic nuances or complex logical relationships [35].

- **Challenges with Long Texts and Multi-hop Reasoning**: For longer documents or complex queries requiring multi-hop reasoning to find an answer, simple averaged embeddings or dot product similarity were often insufficient to effectively capture key information points and their interconnections within the document [36].

2.2.2 Introduction of Reranking and Its Solutions

To address these "coarse-grained" matching and precision limitations, the reranking mechanism was introduced and quickly became an optional yet highly recommended step in the retrieval pipeline [33]. The purpose of reranking is to perform a deeper, more fine-grained secondary evaluation of the Top-K (e.g., top 50) candidate results returned by the initial retrieval phase.

- **Deep Interaction Models**: Unlike the "two-tower" models (where query and document are encoded to embedding separately) often used in the initial retrieval stage, reranking models (e.g., BGE-reranker-base [15]) typically employ a "cross-encoder" architecture. This architecture allows the query and candidate text to be concatenated and processed together through the model. This enables the model to utilize cross-attention mechanisms, performing detailed interactions and comparisons between each token of the query and the document during encoding. This deep interaction can capture more complex and subtle semantic relationships between words, phrases, and contexts, thereby identifying precise matches that initial retrieval might have missed.

- **Higher Precision Discrimination**: Through this deep interaction, reranking models can more accurately assess the true relevance between the query and the candidate text, even distinguishing between texts that are semantically very close but have different underlying meanings. For example, it can more accurately determine that "motor overheating" and "motor overload," despite both containing "motor," refer to fundamentally different types of faults.

- **Precision for Speed Trade-off** : The computational cost of reranking is generally significantly higher than initial retrieval. Therefore, it is applied only to a relatively small subset of candidates (Top-K), achieving the goal of significantly improving final matching precision within acceptable time overheads.

2.2.3 Variants and Future Directions

As the demand for semantic matching continues to deepen, and new challenges emerge in specific application scenarios, relying solely on general embeddings and reranking can no longer meet all needs. To overcome these limitations, a range of variants and enhancement methods have been proposed.

One of them is NER-based Processing. In many specialized domains, specific entities within the text (e.g., product names, fault codes, disease names) carry core semantic information. Traditional embedding methods may struggle to precisely identify and match these entities and their attributes. Therefore, some variant approaches incorporate NER technology [13, 27]. These studies enhance retrieval precision by extracting structured entity information, such as ”equipment name,” ”fault type,” or ”timestamp”, from both the query and candidate texts (i.e., class), and then using these entities as distinct matching features. For instance, if a query is ”fault code B for production line A,” the system not only calculates overall semantic similarity but also explicitly checks if the candidate text mentions ”production line A” and ”fault code B,” improving matching robustness and interpretability. This method is particularly suitable for industrial diagnosis, medical alerts, or legal document retrieval, where precise matching of specific information snippets is crucial. NER outputs can be combined with dense vector embeddings, for example, through fusion or weighted scoring, to achieve more comprehensive matching.

Beyond NER, other advanced approaches have been developed to address different facets of the semantic matching challenge. For instance, Knowledge Graph-enhanced Retrieval integrates Knowledge Graphs (KGs) into the retrieval process to overcome the limitations of general embeddings in handling complex entity relationships and inference [11]. And, Domain Adaptation and Fine-tuning techniques are crucial for improving performance in specialized vertical domains (e.g., healthcare, legal, indutry) where general pre-trained models might fall short due to a lack of domain-specific vocabulary or concepts.

2.3 Classification-Based Methods

Classification-based methods used in semantic matching can be seen as a multi-class classification problem. In this paradigm, each predefined rule or category is treated as a class label [7]. Under this paradigm, each predefined semantic category, intent, or rule is considered as an independent category label. The model receives a user input as input and outputs the most likely matching label. This approach explicit learning of the mapping between inputs and class labels via supervised signals.

2.3.1 Method Evolution and Core Mechanisms

Classification methods for semantic matching have evolved from traditional machine learning to deep learning.

Early Traditional Machine Learning (ML) approaches typically used feature representations like Bag-of-Words or TF-IDF, combined with classifiers such as Naive Bayes, Support Vector Machines (SVM), or logistic regression for semantic matching [20]. These methods were simple and efficient but limited in handling complex semantics and contextual information. While simple and computationally efficient, these early methods struggled profoundly with complex semantics and contextual understanding. They couldn’t grasp the semantic similarity between lexically different words like "car" and "automobile," completely ignored word order (failing to distinguish "dog bites man" from "man bites dog"), couldn't handle polysemy (e.g., "bank" as a financial institution versus a riverbank), and faced the "curse of dimensionality" with high-dimensional, sparse feature spaces [20].

To mitigate these limitations in semantic understanding, shallow neural networks were introduced with more sophisticated feature representations. These methods primarily relied on word embeddings, dense, low-dimensional vectors learned from large corpora (e.g., Word2Vec [21], GloVe [22]) that capture semantic relationships between words. Models could leverage pre-trained word embeddings trained on large corpora, reducing the need for extensive task-specific training data and enabling the capture of semantic similarities between lexically different words. Sequential architectures such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTMs), and Convolutional Neural Networks (CNNs) were applied to process word sequences and learn contextual representations [5]. These networks could learn richer, low-dimensional, dense feature representations from text sequences, allowing them to capture semantic word associations and sequential information. The use of pre-trained word embeddings (such as Word2Vec or GloVe) became standard practice, enabling models to benefit from semantic knowledge learned from large unlabeled corpora. However, shallow neural networks encountered new problems: they still struggled with very long-range dependencies, often forgetting information from earlier parts of the sequence. Crucially, they typically generated a fixed vector representation for each word, failing to handle the context-sensitive nature of polysemous words. Furthermore, they still required substantial amounts of labeled data for effective training from scratch [5].

These persistent challenges paved the way for the era of Pre-trained Language Models (PLMs), notably Transformer-based architectures like BERT [9] and RoBERTa [19]. PLMs revolutionized the field by undergoing large-scale pre-training on vast amounts of unlabeled text, learning incredibly rich linguistic knowledge, syntactic structures, and wide knowledge. For classification tasks, a simple classification head is added atop the pre-trained model, and the entire system is fine-tuned end-to-end using smaller, task-specific labeled datasets. PLMs excel due to their powerful contextual modeling via the self-attention mechanism, which captures complex long-range dependencies and generates dynamic, context-sensitive word representations, effectively resolving the polysemy problem.

Despite these advancements, PLMs in classification-based semantic matching still face inherent limitations.

PLMs still require a non-trivial amount of high-quality labeled data for fine-tuning to achieve expert-level accuracy, especially in highly specialized domains or for tasks involving structurally homogeneous texts where subtle distinctions (e.g., distinguishing "motor overheating" from "motor high temperature") are critical. Generic PLMs may also lack domain-specific knowledge for highly specialized areas, necessitating additional domain-adaptive pre-training or more intensive fine-tuning with in-domain data. The computational resource consumption for both pre-training and fine-tuning PLMs remains substantial, posing a challenge for resource-constrained teams [23].

Furthermore, PLMs struggle with generalizing to entirely new categories unseen during training, typically requiring re-fine-tuning, which limits system flexibility [23]. Finally, the "black-box" nature of PLMs often leads to poor interpretability, which can be a significant drawback in high-stakes domains like healthcare or industrial troubleshooting where transparent decision-making is crucial.

2.4 Related Work

Over the past decade, semantic matching has evolved considerably, with two dominant paradigms emerging: classification-based and retrieval-based approaches. This section reviews the theoretical underpinnings, methodological advances, and domain-specific applications of these two approaches, highlighting their respective advantages and limitations.

2.4.1 Overview of Semantic Matching Paradigms

The choice between retrieval-based and classification-based methods for semantic matching is often dictated by the specific characteristics and constraints of the application. Both paradigms offer distinct advantages and face inherent challenges. The survey [28] on answer generation in QA systems offers a foundational classification of matching strategies, distinguishing among extraction-based, retrieval-based, and generative methods. This work emphasizes that retrieval-based methods excel in scalability and interpretability, whereas classification-based systems often exhibit higher precision in closed-set settings, particularly when fine-tuned on task-specific data.

Retrieval-based methods are suitable for handling dynamic datasets and open-domain candidate sets, where the matching targets aren't fixed, predefined categories but rather an ever-growing or changing collection of documents. Retrieval approaches also demand relatively less labeled data, especially in their initial setup. Pre-trained embedding models can generate text representations without extensive task-specific labeled data for classifier training, making them more feasible in low-resource scenarios. They effectively handle open-ended queries, adapting flexibly to diverse user expressions. Also, adding new content (e.g., a new fault report or product description), only need to simplyly involve updating the vector index, without retraining the entire model, which makes them highly flexible for content updates [14].

However, retrieval methods face notable challenges. Their primary limitation lies in precision for fine-grained semantic distinctions. While reranking [33] can boost accuracy through more complex cross-encoder models [15], distinguishing between highly homogeneous texts with subtle semantic differences (like ”motor overheating” versus ”motor overload”) can still be tough using only similarity calculations. This means they often can not provide explicit classification labels, offering only a similarity rank, which might be insufficient for scenarios requiring definitive decisions. The system's performance also highly depends on embedding quality: if the embeddings don't capture domain-specific semantic nuances, retrieval effectiveness suffers.

In contrast, classification-based methods can effectively differentiate between semantically close texts belonging to different categories, particularly excelling at subtle distinctions within structurally homogeneous texts, assuming sufficient and well-annotated training data is available to capture these differences. Classifiers directly output clear category labels, which is incredibly beneficial for automated decision-making or human intervention.

Despite their power, classification methods have significant drawbacks. Their core challenge is a strong reliance on labeled data. In low-resource scenarios, the scarcity of labeled data severely limits a classification model's performance, often proving to be its

Achilles' heel [12]. Classification methods also struggle with scalability when there are many categories; adding a new category usually necessitates retraining the model, and the training and inference costs climb significantly with more classes. They also cannot handle new, unseen categories from the training set, requiring frequent model updates and re-training, which reduces system flexibility [23]. In some cases, classification models might struggle with distinguishing highly homogeneous texts, as they can't learn clear decision boundaries.

2.4.2 Comparative Studies Across Domains

However, as semantic matching techniques have been extensively applied across diverse domains, it has become evident that different scenarios require significantly different approaches for method selection and optimization, leading to a proliferation of specialized research targeting specific fields and tasks. These domain-specific studies have not only enriched the theoretical framework of semantic matching but also provided more tailored solutions for practical applications.

In the biomedical document processing domain, researchers from Oregon Health & Science University [6] conducted comprehensive comparative analyses between classification and retrieval methods. For classification tasks, the optimal performance was achieved through voting perceptron classifiers combined with chi-square feature selection and binary feature weighting, while for retrieval tasks, the Zettair search engine with optimized Okapi BM25 [25] parameters proved most effective, with keyword-only queries consistently outperforming queries containing all topic words. This comparison revealed fundamental differences: classification methods emphasize feature engineering and model selection, whereas retrieval methods rely more heavily on query optimization and parameter tuning. The comparative study of library classification schemes and information retrieval thesauri [31] further reinforced this distinction, where traditional classification systems emphasize hierarchical organization and standardized categorization, while information retrieval focuses on flexible vocabulary associations and dynamic matching mechanisms. In the chemical compound domain, the contrast becomes even more pronounced [29] : SVM-based classification methods for compound categorization tasks showed optimal performance with extended connectivity fingerprint descriptors, while in ranked retrieval tasks, connected fragment descriptors combined with extended vector-based kernel functions demonstrated superior adaptability. These comparative studies indicate that even when dealing with identical data and similar semantic matching objectives, classification methods tend to achieve deterministic categorization through precise boundary delineation, while retrieval methods prioritize similarity ranking and flexible matching strategies.

Motivated by these cross-domain comparative analyses, this study will systematically conduct a comparative study of retrieval and classification methods for the specific task of matching user descriptions with diagnostic rules in the industrial equipment domain, aiming to explore the performance differences and applicability boundaries of both approaches within this particular application context.

2.5 Measurement

The semantic matching task is evaluated using several widely used metrics that capture ranking accuracy, classification performance, and inference efficiency. Specifically, we report:

2.5.1 Top-k Hit Rate (Top-1 and Top-3 Accuracy)

Top-k hit rate measures the proportion of test instances for which the correct fault label appears in the top k predictions ranked by model confidence or similarity score.

- **Top-1 Hit Rate**: This metric evaluates whether the model's highest-ranked prediction matches the ground truth label. It reflects the model's precision in selecting the single most relevant fault.

$$\text{Top-1 Hit Rate} = \frac{\text{Number of correct Top-1 predictions}}{\text{Total number of samples}}$$

- **Top-3 Hit Rate**: This extends Top-1 by checking whether the true label appears in the top 3 predictions. It provides a more lenient view of performance and is particularly useful in real-world diagnostic scenarios, where multiple plausible fault candidates may need to be suggested.

$$\text{Top-3 Hit Rate} = \frac{\text{Number of samples where true label is in Top-3}}{\text{Total number of samples}}$$

2.5.2 Precision, Recall, and F1 Score

- **Precision**: Precision assesses the proportion of predicted positive cases that are actually correct:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall**: Recall captures the proportion of actual positive cases that were correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1 Score**: F1 Score is the harmonic mean of precision and recall, providing a balanced measure especially useful under class imbalance:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

2.5.3 One-vs-Rest AUC (Area Under the Curve)

To assess the model's discrimination ability across multiple classes, the One-vs-Rest (OvR) AUC strategy is adopted. This involves the following steps:

- **Binary Decomposition**: For each class, a binary classification problem is formed where: 1) The target class is considered the positive class; 2) All other classes are treated as negative.

- **Per-Class AUC Calculation**: AUC is computed individually for each binary classification task, reflecting the model's ability to rank positives higher than negatives for that specific class.

2.5.4 Average Inference Time per Sample

To evaluate the runtime performance of the model, this study measured the *average inference time per sample*. This metric represents the mean amount of time (in milliseconds) required by the model to process a single input instance. It is calculated by dividing the total inference time by the number of samples:

$$\text{Avg inference time (ms)} = \frac{\text{Total inference time (ms)}}{\text{Total number of samples}}$$

All reported Precision, Recall, and F1 scores in this study refer to macro-averaged metrics. This averaging strategy treats all classes equally by computing the metric independently for each class and then taking the unweighted mean.

2.6 Summary

This chapter has established the theoretical and methodological foundations necessary for understanding semantic matching in industrial fault diagnosis contexts. The examination of semantic matching paradigms reveals that the choice between retrieval-based and classification-based approaches is not merely a technical preference but a strategic decision influenced by dataset characteristics, resource constraints, and application requirements.

The technical analysis demonstrates that retrieval-based methods can function with minimal labeled data, leveraging unsupervised or weakly supervised techniques to match queries with relevant historical cases or documentation, which significantly reduces the annotation burden. While, classification-based methods excel in precision-critical applications where fine-grained semantic distinctions must be reliably captured, albeit at the cost of increased data requirements and reduced flexibility.

Domain-specific comparative studies across biomedical, chemical, and library science applications have illuminated a consistent pattern: the optimal approach varies significantly based on the underlying data characteristics and task objectives. These cross-domain insights suggest that industrial fault diagnosis, with its unique combination of technical terminology, hierarchical fault structures, and real-time operational constraints, requires careful empirical evaluation to determine the most suitable semantic matching paradigm.

The measurement framework established in Section 2.5 provides the necessary tools for rigorous comparative analysis, encompassing both accuracy metrics (Top-k hit rates, precision, recall, F1-score) and practical considerations (inference time, scalability). The adoption of macro-averaged metrics ensures fair evaluation across potentially imbalanced fault categories, while the inclusion of ranking-based metrics acknowledges the multi-candidate nature of industrial diagnostic scenarios.

Building upon this theoretical foundation, the subsequent chapters will present a systematic empirical comparison of retrieval-based and classification-based approaches within the specific context of industrial equipment fault diagnosis, thereby contributing to both the semantic matching literature and practical industrial applications.

3 Methodology

3.1 Choice of Research Method

3.1.1 Research Question and Methodological Requirements

This study investigates the comparative performance of retrieval-based and classification-based approaches for semantic matching under challenging real-world conditions, specifically when labeled data is scarce, the label set is fixed, and user-generated texts are highly similar in structure. The target task involves matching free-form user descriptions of equipment faults to structured fault documentation, thereby enabling automated intent recognition for fault diagnosis.

As discussed earlier, the core question is the comparative evaluation of retrieval and classification paradigms under constrained supervision and fixed label sets.

This comparative research question necessitates specific methodological considerations that drive the choice of research approach.

3.1.2 Empirical Method Justification

The research question requires comparative evaluation of two fundamentally different paradigms: similarity-based retrieval and supervised classification. This comparative nature necessitates controlled empirical evaluation where both approaches are implemented under identical conditions and evaluated against consistent performance criteria.

The research adopts a controlled experimental methodology for several reasons:

- **Paradigm Isolation:** Systematic isolation of paradigmatic differences while controlling for confounding variables such as data quality, evaluation metrics, and implementation sophistication enables attribution of performance differences to methodological choices rather than implementation artifacts.
- **Multi-factor Analysis:** The research examines multiple interacting factors beyond the primary paradigm comparison, including semantic enhancement techniques, preprocessing strategies, and input variability effects. The complexity of these interactions necessitates factorial experimental design where individual factor contributions can be isolated through systematic variation.
- **Component-level Understanding:** Modern NLP systems consist of multiple modular components that can be combined in various configurations. Understanding which specific enhancement techniques contribute meaningfully to overall system performance requires systematic evaluation of individual components.

3.1.3 Performance Operationalization

Performance is operationalized as the accuracy of semantic matching between user descriptions and structured fault documentation, evaluated through a unified set of metrics applied to both paradigms. Specifically, the evaluation includes top-k accuracy, precision, recall, F1 score, and One-vs-Rest AUC, providing a comprehensive view of both ranking quality and classification performance. Additionally, the average inference time per sample is measured to assess the computational efficiency of each method under deployment-like conditions.

To ensure robustness, performance is further evaluated across systematically varied input styles (including vague, precise, and overly verbose descriptions), that reflect the linguistic diversity encountered in real-world user interactions. This robustness assessment helps bridge the gap between controlled experimental settings and practical deployment scenarios, highlighting the system's ability to generalize under realistic usage condition.

3.2 Technical Components and Enhancement Strategies

3.2.1 Paradigm Overview

To better illustrate the methodological differences and overall design, this study provides a comparison between classification-based and retrieval-based paradigms (Figure 1), as well as a detailed pipeline of the retrieval-based approach adopted in our system (Figure 2). The target application domain involves matching free-form user descriptions of equipment faults to structured fault documentation, enabling automated intent recognition for industrial fault diagnosis systems.

Figure 1 presents an overview of the two paradigms.

In the classification-based method, a supervised learning pipeline is adopted that relies on specific feature extraction and representation strategies. During the training phase, labeled pairs of user descriptions and fault rule labels are processed through feature extraction mechanisms. The feature extraction process involves tokenizing the user description, converting tokens to their corresponding embeddings, and often applying pooling strategies (mean, max, or attention-based pooling) to obtain a fixed-size representation. This representation is then passed through a classification head (typically fully connected layers with softmax activation) to produce probability distributions over fault rule labels. During inference, a new user description undergoes the same feature extraction pipeline, tokenization, embedding lookup or contextual encoding, pooling, and classification—to produce a predicted fault rule label. This approach treats the task as a multi-class classification problem, where each possible fault rule corresponds to a distinct label, and the model learns direct mappings from text features to categorical outputs.

In contrast, the retrieval-based method formulates the problem as a similarity search task. Both the user description and fault rule descriptions are encoded into a shared semantic vector space using pre-trained language models or fine-tuned encoders. The system computes pairwise similarity scores between the embedded user description and each embedded fault rule. The most relevant fault rule is then selected based on the highest similarity score.

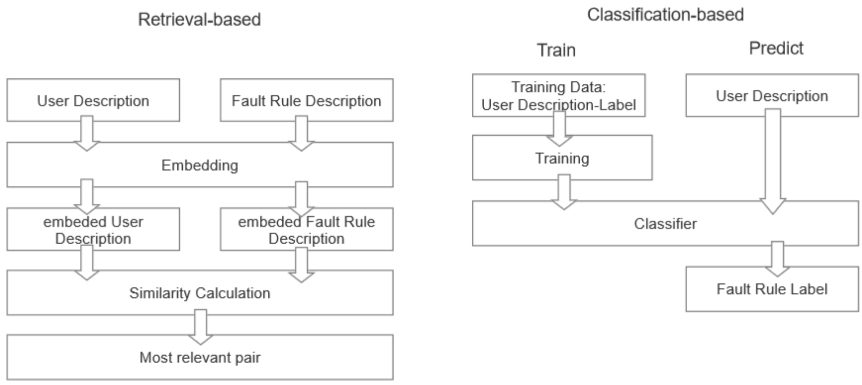

Figure 1: Architectural distinctions between class-based and retrieval-based approaches

Figure 2 further breaks down the retrieval-based method into its core process: Retrieval and Reranking.

In the retrieval stage, both the user description and fault rule descriptions may undergo optional preprocessing steps, such as NER, which are marked with dashed boxes in the diagram. These components are not mandatory, but can help extract key entities and improve embedding quality by focusing on salient terms.

Subsequently, the texts are encoded to dense vector representations. A similarity calculation (e.g., cosine similarity) is performed between the embedded user description and each embedded fault rule, resulting in a list of top-K most similar candidates. The reranking stage, also indicated as an optional module with dashed boxes, can be applied to further refine these Top-K candidates. A reranker model, such as a cross-encoder or a task-specific scoring function, re-evaluates the retrieved pairs and produces a final relevance score. The candidate with the highest reranked score is then selected as the final match.

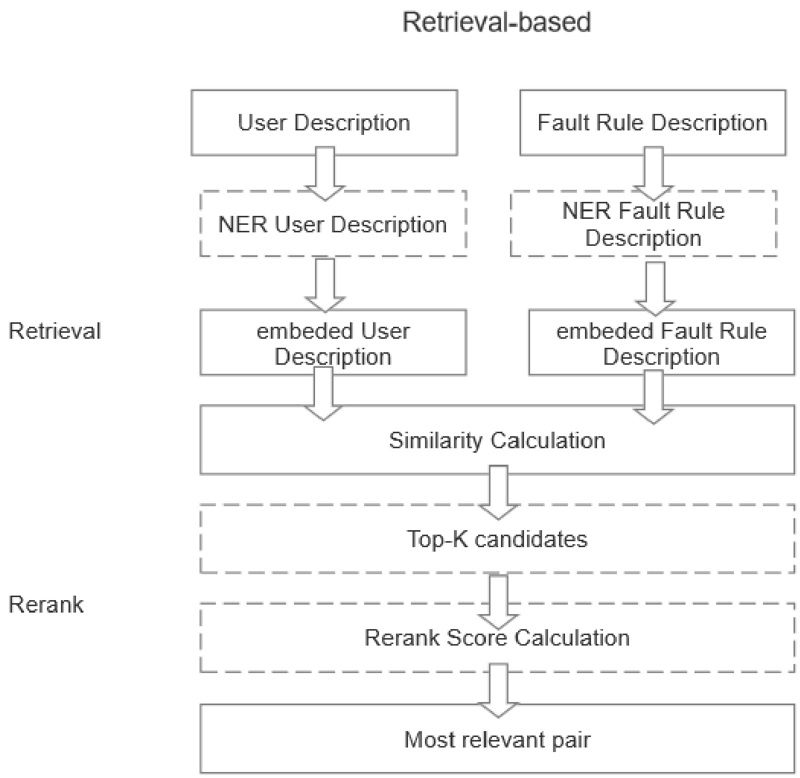

Figure 2: Internal structure of the retrieval-based method

3.2.2 Base Models and Architectures

The experimental framework employs four core models, each serving distinct functions within the semantic matching pipeline:

bge-base-en : A dense retrieval-optimized embedding model, capable of generating high-quality text representations suitable for semantic similarity tasks [32][3] [16].

bge-reranker-base : A reranking model designed for semantic comparison, providing fine-grained judgment on Top-K candidate matches following initial retrieval [15][4][16].

sci-bert : A BERT variant trained on scientific and medical corpora, particularly effective for NER in domain-specific texts [1].

bert-base-uncased : A general-purpose BERT model well-suited for fine-tuning in multi-class classification tasks, with strong capacity for modeling contextual relationships [9].

3.2.3 NER Enhancement Strategies

In preprocessing pipeline, NER is used to enrich the semantic content of the text data. This study specifically explored three distinct NER-based strategies. The specific processing methods for each strategy are detailed below.

Method 1: Reorganize This strategy aims to transform the often fragmented entity and state information from the original text into a more structured and concise 'entity is state' declarative phrase. Detailed Steps:

1. **NER Extraction:** First, using an NER model to identify all relevant named entities (e.g., equipment components, system indicators) and their corresponding states (e.g., operational status, fault manifestations) from the original text description. These are typically extracted as [entity,state] pairs.

2. **Phrase Construction:** For each extracted [entity,state] pair, this study reconstructs it into a new phrase following the 'entity is state' format (e.g., unit is turned off").

3. **Text Reorganization:** All constructed 'entity is state' phrases are then concatenated, usually separated by commas or semicolons, to form the new processed text. The remaining parts of the original text are discarded, keeping only the extracted core information.

Method 2: Keyword Augmentation This strategy enhances semantic information by appending the NER-extracted entities and states as additional keywords to the end of the original text, preserving the original text's integrity. This helps models focus on critical concepts within the text while still retaining the original context.Detailed Steps:

1. **NER Extraction:** Similar to the Reorganize" strategy, identifying entities and their corresponding states from the original user description.

2. **Keyword List Generation:** All extracted entities and states (e.g., "unit", "turned off", "red light", "on", etc.) are flattened into a continuous list of keywords.

3. **Text Appending:** Finally append these generated keywords to the end of the original description.

Method 3: XML-tagged Format This strategy involves embedding XML-like tags directly within the original text to explicitly mark identified entities and states. This allows models to leverage both the raw textual context and the structured entity information simultaneously. Detailed Steps:

1. **NER Extraction and Span Localization:** NER is performed to not only extract the content of entities and states but also to precisely locate their start and end positions within the original sentence.

2. **Tag Insertion:** Based on the type of entity or state (e.g., <ENTITY> for entities, <STATE> for states), corresponding opening (e.g., <ENTITY>, <STATE>) and closing (e.g., </ENTITY>, </STATE>) XML tags are inserted at the identified text spans within the original description. The rest of the untagged text remains unchanged.

3.2.4 Application Scope Variations

Each NER enhancement strategy can be applied across three different scopes, creating systematic variations in preprocessing coverage:

- **User Description Only:** NER enhancement is applied exclusively to user-generated fault descriptions, while fault rule documentation remains in its original form. This asymmetric approach focuses enhancement efforts on the more variable and potentially less structured user input.
- **Rule Description Only:** NER enhancement targets only the structured fault rule documentation, leaving user descriptions unmodified. This approach assumes that enhancing the more structured, authoritative documentation may improve matching against variable user inputs.
- **Both Descriptions:** NER enhancement is applied symmetrically to both user descriptions and fault rule documentation, creating consistent entity-enhanced representations across both sides of the matching task. This comprehensive approach maximizes entity-based semantic alignment.

3.3 Application of Research Method

3.3.1 Data Collection

The task is framed as a semantic alignment problem between two textual sources: user descriptions and fault rule descriptions. Both sources are composed in English. The goal is to correctly match a given user description to its corresponding fault rule.

To evaluate different approaches, this study conducted comparative experiments between retrieval-based and classification-based methods. The retrieval-based method operates in an unsupervised manner: no model training is performed during the experiments. Instead, it directly computes similarity scores between user descriptions and fault rule descriptions to determine the best match. This approach relies purely on semantic similarity measures such as embedding distances.

In contrast, the classification-based method is supervised. It takes user description texts as input and outputs a predicted fault rule label. The classifier is trained on labeled pairs of user descriptions and their corresponding fault rules, learning to map textual inputs to discrete fault rule categories.

By comparing these two approaches, this study aims to assess which method better captures the semantic correspondence between user input and fault rules under varying linguistic styles and data conditions.

1) User Description data User descriptions were generated by prompting Large Language Models (LLM) (ChatGPT) with Munters AB company documentation and predefined fault rule description sets. For each rule, three styles of user inputs were created to reflect different levels of linguistic clarity: categorized as Vague, Precise, and Lengthy styles. Specifically, the generation process involved providing the LLM with usage scenarios and background context related to a particular product, such as a fault diagnosis QA system. For each fault rule, a base description was supplied as the seed input. The LLM then generated corresponding user descriptions based on this base description, simulating how different users might describe the same fault in diverse linguistic styles. Through this approach, this study established mappings between multiple user input variants and their corresponding fault rules.

The resulting dataset comprises 8,195 instances, with anonymized field names to ensure confidentiality. Each of the 17 categories contains approximately 400 to 500 samples, encompassing three distinct linguistic styles to reflect diverse expressions within the same class. The relevant fields used in this study include:

- **Style**: Indicates the linguistic tone or expressiveness of the user description (e.g., Vague, Precise, Lengthy).
- **Description**: Free-text user descriptions of faults related to industrial equipment.
- **Label**: The annotated fault class, represented as an integer (e.g., 1, 2, 3, ..., 17), mapping to a specific structured fault description. There are 17 classes in total.

Explaination about the 3 linguistic styles:

- Precise descriptions: Technical, clear, and concise expressions that may include model numbers, error codes, sensor values, and suspected causes.
- Vague descriptions: Non-technical and unclear expressions that typically describe only symptoms or express confusion with minimal detail.
- Lengthy descriptions: Overly detailed or anecdotal expressions that may include irrelevant background information, speculation, or personal opinions.

2) Fault Rule Description data Fault descriptions originate from structured internal documents, 17 in total. They have been anonymized for privacy. The key fields used include:

- **Fault Indicator**: A textual representation of observable system behavior or alert messages, such as “Failure to Start” or “Abnormal Temperature shown in the screen”.
- **Label**: The categorical fault label consistent with that used for user descriptions, from 1 to 17.

3.3.2 Data Processing

1) User Description data Although user descriptions are intended to simulate real-world user input, the majority were generated by a large language model (LLM) based on internal documentation and predefined fault rules. These AI-generated texts are inherently well-formed, free from typographical errors and formatting inconsistencies, and can thus be regarded as clean input.

In this study, the primary preprocessing step focuses on the application of NER to extract meaningful, domain-specific entities. Specifically, NER was selectively applied to both user descriptions and fault rule texts to identify key elements relevant to industrial fault diagnosis, such as system components, operational states, numerical parameters, and performance indicators. These extracted entities were then integrated into the data representations to improve semantic precision and model interpretability.

Three distinct NER-based strategies were explored as optional representation enhancement techniques. In the subsequent experiments, some settings utilize the original (unprocessed) texts, while others employ samples enhanced with NER-derived entity information. An illustrative example of entity extraction and its incorporation is presented in table 1.

Table 1: Examples of Different NER-based enhancement strategies

Field	Explanation	Example
Original Description	Raw user-reported issue description	It turned off with a red light, and the screen has some kind of sensor fault warning.
NER and Pairs	Extracted pairs of entities and their associated states	[[unit, turned off], [red light, on], [screen, sensor fault warning]]
Reorganize	Reformatted as 'entity is state' expressions	unit is turned off, red light is on, screen is sensor fault warning.
Keyword	Appends NER results as keywords for emphasis	It turned off with a red light, and the screen has some kind of sensor fault warning. Keywords: unit, turned off, red light, on, screen, sensor fault warning.
XMLTagged	XML-tagged format for structured parsing	It <STATE> turned off </STATE> with a <ENTITY> red light </ENTITY>, and the <ENTITY> screen </ENTITY> has some kind of <STATE> sensor fault warning </STATE>.

2) Fault Rule Description data Fault rule descriptions, authored by domain experts, are linguistically consistent and technically accurate. Consequently, no additional textual cleaning or normalization is required. To maintain alignment with the preprocessing pipeline applied to user inputs, the same three NER-based strategies were optionally applied to the fault rule texts as well. This ensures a consistent representation framework across both input types.

3.3.3 Parameters and System Configuration

This section outlines the parameter settings, model configurations, and system setup used in our experiments, which consist of three main pipelines: BGE-based retrieval, reranker-enhanced retrieval, and BERT-based multi-class classification.

1) BGE Embedding for Vector-Based Retrieval This study employed the bge-base-en model to generate semantic embeddings for both query and rule descriptions. Tokenization was handled via the HuggingFace AutoTokenizer. Related parameters and settings can be seen as followd:

Parameter	Value
model_name	BAAI/bge-base-en
tokenizer	AutoTokenizer.from_pretrained
query_prefix	"query: "
max_length	512
padding	True
truncation	True
pooling_strategy	"mean"
inference_mode	model.eval(), torch.no_grad()
embedding_dim	768

Table 2: BGE embedder parameter settings

For better alignment in search contexts, each query was prepended with the string 'query:' before encoding. Embeddings were derived using mean pooling over the final hidden state:

$$\text{Embedding} = \frac{\sum(\text{HiddenStates} \cdot \text{AttentionMask})}{\sum(\text{AttentionMask})}$$

Cosine similarity was then computed between each user description and the predefined rule embeddings. The top-3 most similar rules were selected as predictions.

2) BGE Reranker To further improve retrieval accuracy, this study implemented a two-stage retrieval pipeline:

- **Top-K Vector Search**: The top 5 candidates were retrieved using cosine similarity on BGE embeddings.
- **Reranking**: this study used the BAAI/bge-reranker-base model, a sequence classification transformer, to score the relevance between the query and each of the 5 candidates. The top-3 highest scoring labels were retained as final predictions.

The reranker model outputs scalar relevance scores, and was used in inference mode without additional fine-tuning.

3) BERT based multi-class Classification A multi-class classification model based on bert-base-uncased was trained for direct prediction of issue categories. The model architecture consists of:

Component	Description
Encoder	A pre-trained BERT encoder
Dropout	A dropout layer (probability = 0.1)
Output Layer	A fully-connected layer (output nodes = 17)

Table 3: Architecture components of the classification model

The training setup is summarized as follows:

Parameter	Value
Loss Function	Binary Cross-Entropy with Logits
Batch Size	16
Epochs	3
Learning Rate	2×10^{-5}
Optimizer	AdamW
Train/Test Split	80% / 20%

Table 4: Training configuration details

Evaluation metrics included top-1 accuracy, top-3 accuracy, precision, recall, and macro F1-score.

4) System Configuration and Runtime Logging: Experiments were executed on the following hardware and software setup:

Component	Specification
GPU	NVIDIA GRID V100D-16Q (CUDA Capability 7.0)
CPU	Intel64 Family 6 Model 85 Stepping 7
RAM	42 GB
OS	Windows 10 (64-bit)

Table 5: Hardware and system specifications

3.4 Experimental Design Framework

3.4.1 Multi-Factorial Experimental Structure

The methodology incorporates multi-factorial experimental design to address the research question's inherent complexity. Three primary factors are systematically varied:

1. **Paradigmatic Approach:** Retrieval-based versus classification-based methods
2. **Semantic Enhancement Strategies:** Baseline, NER preprocessing (different NER strategies), reranking, combined approaches
3. **Input Characteristics:** Linguistic style variations (vague, precise, lengthy)

This design enables examination of main effects, interaction effects, and conditional performance patterns that single-factor experiments cannot reveal.

3.4.2 Ablation Study Integration

Within the experimental framework, ablation studies decompose complex system performance into constituent component contributions. This approach systematically evaluates individual components (NER, cross-encoder reranking) by comparing performance with and without each component while holding other factors constant.

The ablation methodology is particularly warranted given the modular nature of modern NLP systems, providing granular insight into component-level effectiveness and enabling informed architectural decisions for practical implementations.

3.4.3 Experimental Setting

To assess the effectiveness of different semantic enrichment strategies in fault classification, this study designed five experimental settings. Each setting represents a distinct combination of representation, retrieval, and classification techniques. Details of the representation, retrieval, and classification components used in each of the five experimental settings are presented in Table 6.

These settings are summarized in the accompanying table and described below.

Experiment 1 (Exp1) serves as the baseline configuration. In this setting, this experiment uses the original user and fault rule texts without any preprocessing or transformation. Semantic similarity between texts is computed using sentence embeddings generated by the bge-base model. No NER is applied, and no classification model is used.

Experiment 2 (Exp2) extends the baseline by introducing a reranking step. The initial retrieval is still based on bge-base embeddings, but the retrieved candidates are further reranked using the bge-reranker-base model. This allows for more refined matching based on contextual similarity, while the original texts remain unmodified and no classification model is involved.

Experiment 3 (Exp3) introduces NER as a preprocessing step. Specifically, both user descriptions and fault rule texts are rewritten to highlight domain-specific entities, such as equipment components, operating states, and numerical values, by using the sci-bert NER model. These rewritten texts are then compared via embedding similarity using the bge-base model. This configuration excludes reranking and classification.

Experiment 4 (Exp4) combines NER-based rewriting with both embedding similarity and reranking. The rewritten texts are embedded using the bge-base-en model, and reranking is again performed with bge-reranker-base. This setting tests the cumulative effect of entity-focused preprocessing and reranking on retrieval accuracy.

Experiment 5 (Exp5) takes a different approach by discarding the retrieval framework entirely. Instead, this setting treats fault classification as a direct text classification task. The user description is input to a fine-tuned bert-base-uncased model, which predicts the fault label without comparing it to any fault rule text.

Table 6: Comparison of Retrieval and Classification Methods with Different Enhancements

Exp	Method	Enhancements
1	Retrieval	None
2	Retrieval	+Reranker
3	Retrieval	+NER
4	Retrieval	+NER +Reranker
5	Classification	Supervised Learning (Train/Test Split 80%/20%)

Our experiments involve three distinct phases, each utilizing different model components:

- **Retrieval Phase**: Uses embedding models for dense retrieval
- **Reranking Phase**: Applies cross-encoders to refine retrieval results

- **Classification Phase**: Performs final prediction using supervised learning

Details of the models used in each component are provided in Table 7.

Table 7: Models Used in Each Phase

Model	**Purpose**
bge-base-en	Dense retriever (Retrieval Phase)
bge-reranker-base	Reranker (Reranking Phase)
sci-bert	Domain-specific NER (Preprocessing)
bert-base-uncased	Classifier (Classification Phase)

Detailed Configuration of Exp3: Exp3 is designed to systematically explore the effect of different NER integration strategies. It is subdivided into a total of 9 configurations, derived from a 3×3 matrix combining:

- **NER enhancement methods**: Reorganize, Keyword Augmentation, and XML-tagged Format;
- **Application scopes**: Applying NER to the user description only, the rule description only, or to both sides.

The detailed setup of the nine configurations in Exp3 is summarized in table below.

Table 8: The 3×3 Experimental Setup in Exp3

NER Strategy	NER on User Description Only	NER on Rule Description Only	NER on Both Descriptions
Reorganize	Exp3.1	Exp3.2	Exp3.3
Keyword Augmentation	Exp3.4	Exp3.5	Exp3.6
XML-tagged Format	Exp3.7	Exp3.8	Exp3.9

Detailed Configuration of Exp4: The goal of Exp3 is to identify the most effective way of incorporating NER information, both in terms of the embedding format and the optimal point of application.

Exp4 builds upon the findings of Exp3. Instead of repeating all 9 configurations, Exp4 uses only the best-performing NER setup from Exp3 to evaluate the added value of a reranking model. The comparison between Exp3 and Exp4 is focused not on the effectiveness of NER itself, but rather on how reranking can further enhance performance under an already optimized NER setting.

3.4.4 Analytical Dimensions

The experiments are analyzed along several key analytical dimensions. Through all these comparisons, the goal is to quantify the optimal use of domain-aware NER techniques for enhancing semantic alignment between user descriptions and structured fault rule texts in a domain-specific retrieval setting. Moreover, the results allow us to assess the added value and overall effectiveness of incorporating NER-based preprocessing within this industrial diagnostic context.

1) Retrieval-Based versus Classification-Based Approaches This dimension addresses a central research question: In scenarios characterized by limited training resources, a fixed label set, and structurally similar textual inputs, how do retrieval-based methods compare to classification models in terms of both accuracy and efficiency?

Specifically, this study contrasts Exp1 through Exp4, which rely on various retrieval-based configurations, with Exp 5, which employs a direct text classification model. This comparison allows us to assess the trade-offs between lightweight retrieval architectures and data-intensive classification models in the context of fault diagnosis.

2) Effectiveness of Semantic Reranking To evaluate the contribution of the semantic reranking component, tihs study conducts pairwise comparisons between experiments with and without reranking. In particular:

- Exp1 versus Exp2 assesses the impact of reranking on original, unprocessed texts.
- Exp3 versus Exp4 evaluates reranking effectiveness when applied to entity-enhanced (NER-enchanced) texts.

This analysis isolates the effect of the reranker in refining the retrieval results beyond raw embedding similarity.

3) Effectiveness of NER-Based Preprocessing This dimension investigates the role of domain-specific NER in enhancing semantic understanding through preprocessing. This study conducts this analysis in two stages.

The first stage focuses on identifying the optimal strategy for applying NER-based sentence rewriting. To this end, this study examined 9 sub-experiments under the Exp3 framework, each corresponding to a different NER configuration. These variations involve applying NER to the user descriptions, the fault rule descriptions, or both, and differ in the ways extracted entities are integrated back into the text. This comparison aims to answer two key questions:

- Which sentence enhancement strategy based on NER leads to the highest improvement in retrieval performance?
- In the current semantic retrieval setting, is NER-based rewriting more effective when applied to both user descriptions and fault rule descriptions, or only to one side?

The second stage evaluates the added value of NER-based rewriting by comparing it with baseline methods that do not incorporate any entity-level processing. Specifically, this study assesses whether integrating NER enhances retrieval quality beyond what is achievable using original texts and semantic similarity alone. This study performs two direct comparisons for this purpose.

- Exp1 versus Exp3: Evaluates whether incorporating NER-based rewriting improves retrieval quality when used with basic embedding similarity.
- Exp2 versus Exp4: Assesses whether the integration of NER-based preprocessing remains beneficial when combined with semantic reranking.

Through all these comparisons, the goal is to quantify the optimal use of domain-aware NER techniques for enhancing semantic alignment between user descriptions and structured fault rule texts in a domain-specific retrieval setting. Moreover, the results allow us to assess the added value and overall effectiveness of incorporating NER-based preprocessing within this industrial diagnostic context.

4) Effective of Input User Description This analysis examines how different user input types (lengthy, precise, and vague) affect model performance. The goal is to assess system robustness under varying input formulations, which commonly occur in real-world settings. By comparing model behaviors across these input types, this study aims to identify which methods are more resilient to input variability, and where failure modes occur. This informs design choices for systems deployed in low-resource, fixed-category tasks, where user inputs may lack consistency or detail.

4 Results and Analysis

This section presents a comprehensive evaluation of the proposed model's performance across different experimental settings. The analysis is organized into four parts. First, this study provides an overall performance assessment on the entire test set, including key metrics such as accuracy, precision, recall, F1-score, and AUC.

Second, this study compares the performance variations brought by different NER methods.

Third, this study investigates how the model performs on different types of input samples, specifically focusing on precise, vague, and lengthy descriptions to evaluate its robustness against linguistic diversity.

Finally, this study analyze the model's predictive performance at the label level to identify the differences in the effectiveness of experiments and models on different error categories.

4.1 Overall Performance

Table 9 presents the overall performance of all five experimental configurations described in Method Section. To ensure fair comparison, all experiments were conducted on the same dataset and evaluated using consistent metrics, including Top-1/Top-3 Accuracy, Precision, Recall, F1 score, Macro-AUC, and average inference time.

For Exp3, which includes a 3×3 grid of configurations, this study reports the performance of its best-performing sub-experiment. The best coonfiguration is: using Keyword Augmentation applied to the user description only, as identified in Experimental design section.

The performance comparison reveals Exp5 (Classification-based) achieves the best overall performance, with near-perfect scores across all evaluation metrics. It reaches a Top-1 Accuracy of 0.9939, F1 score of 0.9940, and ROC AUC of 0.9980. However, this comes at the cost of relatively high inference time (3.18 ms), which may affect real-time deployment scenarios. Figure 3 presents the training loss curve for the classification model, indicating stable convergence and effective learning throughout the training process.

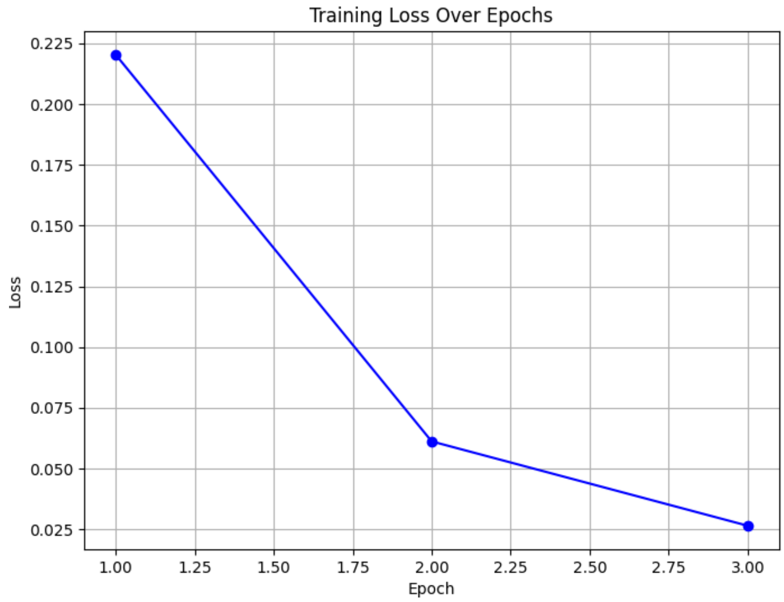

Figure 3: Training loss curve of the classification-based model

As for the Retrieval-based methods, they are inferior to classification methods in the current scenario. Exp2 (Retrieval with reranking) achieves strong performance across all metrics, including a Top-1 Accuracy of 0.7413 and F1 score of 0.7358, demonstrating the value of adding a reranker on top of a standard embedding-based retrieval pipeline. The trade-off, however, lies in its high average inference time (313.21 ms) due to reranking complexity.

Exp3 (best NER-enhanced retrieval configuration) provides a compelling middle ground: a Top-1 Accuracy of 0.4855, relatively high Precision (0.6481), and the lowest inference time among all setups (0.38 ms). This highlights the practical efficiency of targeted NER augmentation without reranking.

Exp4 builds directly upon the best setup from Exp3, integrating a reranker on top of the Keyword-Augmented input. As expected, it boosts Top-1 Accuracy to 0.7097 and improves F1 to 0.7080, demonstrating that reranking adds substantial value when applied to a semantically enriched input. However, it also significantly increases inference time (400.69 ms), the highest among all retrieval-based setups.

Exp1 (baseline), which uses raw text and basic embedding similarity, performs the worst across all metrics (e.g., Top-1 Accuracy: 0.5269, F1: 0.5204), underscoring the limitations of relying solely on naive similarity without augmentation or reranking.

Table 9: Performance metrics across different experimental setups.

Exp	Top1-Acc	Top3-Acc	Precision	Recall	F1	AUC	AvgInferTime(ms)
1	0.5269	0.7352	0.6373	0.5282	0.5204	0.8334	0.39
2	0.7413	0.8149	0.7931	0.7418	0.7358	0.8896	313.21
3	0.4855	0.7505	0.6481	0.4853	0.4851	0.8357	**0.38**
4	0.7097	0.7468	0.7506	0.7140	0.7080	0.8950	400.69
5	**0.9939**	**0.9969**	**0.9940**	**0.9940**	**0.9940**	**0.9980**	3.18

4.2 Performance by NER methods

Table 10 presents the results of nine configurations from Exp3, evaluating three NER tagging methods across different application scopes.

Table 10: Performance Comparison across NER Tagging Methods and Application Scope

Tagging	**Application Scope**	Top1 Acc	Top3 Acc	Precision	Recall	F1	AUC
Keyword	NER User Description / Original Rule	**0.4855**	**0.7505**	**0.6481**	**0.4853**	**0.4851**	**0.8357**
	NER User Description / NER Rule	0.2222	0.3487	0.1794	0.2149	0.1420	0.6192
	Original User Description / NER Rule	0.2222	0.3487	0.1794	0.2149	0.1427	0.6192
Reorganization	NER User Description / Original Rule	**0.4754**	**0.7468**	**0.5604**	**0.4784**	**0.4699**	**0.8283**
	NER User Description / NER Rule	0.2045	0.3714	0.1643	0.1992	0.1398	0.6202
	Original User Description / NER Rule	0.2245	0.3196	0.1854	0.2180	0.1434	0.6145
XML	NER User Description / Original Rule	**0.4594**	**0.6316**	**0.5955**	**0.4590**	**0.4474**	**0.7745**
	NER User Description / NER Rule	0.2177	0.3447	0.1410	0.2106	0.1303	0.6156
	Original User Description / NER Rule	0.2177	0.3447	0.1410	0.2106	0.1303	0.6156

The findings are summarized as follows:

Among all configurations, the combination of Keyword Augmentation applied only to the user description yields the best performance in this dataset across all evaluation metrics, including Top-1 Accuracy (0.4855), Top-3 Accuracy (0.7505), Precision (0.6481), Recall (0.4853), F1 score (0.4851), and AUC (0.8357).

For Comparison of NER Tagging Methods, among the three NER tagging strategies (Keyword Augmentation, Reorganization, and XML-tagged Format), Keyword Augmentation demonstrates the most effective performance overall in this dataset. When applied to the user description only, it achieves the highest scores across all metrics, including the best AUC of 0.8357, indicating superior discriminative ability. In comparison, Reorganization and XML-tagged Format under the same input conditions yield lower performance across all metrics, with AUC scores of 0.8283 and 0.7745 respectively, alongside substantially lower accuracy and F1 scores.

For Effect of NER Application Scope, across all three tagging strategies, the best performance is consistently observed when NER is applied only to the user description, while the rule description remains unaltered. Applying NER to both the user and rule descriptions, or only to the rule description, significantly reduces effectiveness across all metrics. For example, under the Keyword method, not only does Top-1 Accuracy drop dramatically from 0.4855 (user-only) to 0.2222 (both sides), but AUC also decreases substantially from 0.8357 to 0.6192, indicating a significant loss in the model's ability to distinguish between relevant and irrelevant matches. This suggests that overprocessing or altering both sides may introduce noise or harm alignment, while focusing preprocessing on the user side retains clarity and maximizes the value of NER enhancements.

Combining the two findings above, the Keyword Augmentation method applied exclusively to the user description is confirmed as the best-performing configuration in this dataset. This setup outperforms all others by a significant margin across all evaluation metrics, including achieving the highest AUC of 0.8357, demonstrating superior ranking quality and discriminative performance. This configuration is selected as the foundation for Exp4, which will evaluate the potential performance gain from incorporating a reranking stage on top of this optimized NER setup.

4.3 Performance by Input Description Type

Table 11 provides a detailed breakdown of model performance across three distinct types of user input descriptions: lengthy, precise, and vague. This categorization reflects common linguistic patterns observed in real-world user queries and allows us to evaluate the robustness of each experimental configuration in handling various expression styles.

Table 11: Performance Breakdown by Input Type Across Experiments

Exp	**Type**	Top-1 Accuracy	Top-3 Accuracy	Precision	Recall	F1-score	**AUC**
1	Lengthy	**0.7545**	**0.8651**	**0.7623**	**0.7545**	**0.7223**	**0.9204**
	Precise	0.3365	0.5526	0.2102	0.3365	0.2163	0.7499
	Vague	0.5568	0.8641	0.5741	0.5537	0.5203	0.8910
2	Lengthy	**0.9820**	**1.0000**	**0.9844**	**0.9820**	**0.9819**	**0.9982**
	Precise	0.6515	0.6782	0.6391	0.6515	0.5690	0.8251
	Vague	0.6040	0.8116	0.6432	0.6049	0.5706	0.8702
3	Lengthy	**0.7647**	**0.9106**	**0.8029**	**0.7647**	**0.7303**	**0.9434**
	Precise	0.2371	0.5668	0.1204	0.2371	0.0996	0.7487
	Vague	0.5448	0.8468	0.5859	0.5403	0.5168	0.8790
4	Lengthy	**0.8616**	**0.8706**	**0.8887**	**0.8616**	**0.8597**	0.9224
	Precise	0.6709	0.8585	0.7593	0.6709	0.6446	0.8999
	Vague	0.5960	0.7933	0.5983	0.5923	0.5550	0.8549
5	Lengthy	0.9943	0.9962	0.9934	0.9939	0.9937	0.9975
	Precise	0.9926	0.9970	0.9931	0.9923	0.9927	0.9979
	Vague	**0.9954**	**0.9977**	**0.9954**	**0.9972**	**0.9962**	**0.9989**

Across the board, this study observes that user input type has a significant impact on model effectiveness.

Across all retrieval-based configurations (Exp1 – Exp4), lengthy user descriptions consistently yield the best performance, often with significantly higher precision and recall compared to precise or vague inputs. Precise inputs, despite their apparent clarity, result in the lowest performance across nearly all metrics. This trend suggests a potential gap between concise user language and the broader rule representations in the dataset. Vague inputs perform moderately well, often surpassing precise inputs, but consistently underperform compared to lengthy descriptions.

Classification-based approach (Exp5) shows high robustness across all input types, with nearly uniform performance exceeding 0.99 on all metrics.

4.4 Per-Class Performance

To gain deeper insight into class-wise discriminative performance, this study reports the per-class ROC AUC across all five experimental settings in Table 12.

Table 12: Per-class ROC AUC Performance Across Experiments (Exp1–Exp5)

Class	Exp1	Exp2	Exp3	Exp4	Exp5
1	0.5708	0.7450	0.5708	0.9846	0.9982
2	0.7354	0.8350	0.7354	0.8863	1.0000
3	0.7891	0.7689	0.7891	0.8958	1.0000
4	0.7843	0.8342	0.7843	0.6566	0.9990
5	0.8927	0.9519	0.8927	0.9130	0.9990
6	0.8493	0.9508	0.8493	0.9763	0.9883
7	0.6639	0.9056	0.6639	0.9122	1.0000
8	0.9268	0.9697	0.9268	0.9739	1.0000
9	0.9244	0.9193	0.9244	0.9232	1.0000
10	0.8537	0.9773	0.8537	0.9523	1.0000
11	0.9718	0.9567	0.9718	0.7180	0.9995
12	0.9436	0.9859	0.9436	0.8698	1.0000
13	0.7965	0.8583	0.7965	0.9221	1.0000
14	0.7728	0.7391	0.7728	0.9342	0.9882
15	0.8189	0.7470	0.8189	0.8676	0.9935
16	0.9603	0.9809	0.9603	0.9077	1.0000
17	0.9531	0.9978	0.9531	0.9216	1.0000

Per-Class Trends Across Experiments

Exp5 consistently achieves near-perfect ROC AUC scores (≥0.99) across all 17 classes, confirming the superior class-separability of the fine-tuned classification-based approach.

Exp4, while still retrieval-based, shows substantial gains in ROC AUC compared to Exp 1–3. Several classes (e.g., Class 1: 0.9846, Class 8: 0.9739) approach the classification model's performance, indicating the effectiveness of reranking in enhancing semantic matching.

Exp1, which lacks reranking or fine-tuning, demonstrates the weakest discriminative ability, with some classes falling below 0.60. This indicates significant overlap or misalignment between user descriptions and candidate representations in the absence of enhancement mechanisms.

Class-wise variability is also evident in lower-performing setups. For example, Class 4 exhibits a significant drop in Exp4 (0.6566) despite better performance in other classes, highlighting the need for further balancing or error correction in reranking logic for specific categories.

Inter-Class Performance Comparison

Across all five experiments, Classes 1 and 2 show consistently lower ROC AUC scores compared to other classes. In particular, Class 1 achieves a score of only 0.5708 in Exp1 and Exp3, which is the lowest among all classes. In contrast, Classes 8–12 and 16–17 consistently achieve high scores across all experiments, often exceeding 0.95.

Specifically, Classes 8 and 9 exhibit high baseline performance in Exp1 (0.9268 and 0.9244, respectively) and reach perfect classification (1.0000) by Exp5. Classes 16 and 17 also start strong (0.9603 and 0.9531) and achieve perfect scores in Exp5. This indicates that these fault types are easier to discriminate from others across all experimental conditions.

4.5 Discussion

This section presents a comprehensive discussion of the experimental results, organized around four key analytical dimensions.

Under the constraints of a low-resource environment, a fixed set of fault categories, and structurally homogeneous textual inputs, firstly compare retrieval-based approaches with classification-based methods to examine trade-offs between data efficiency and predictive accuracy. Secondly, this study assesses the effectiveness of a semantic reranking module in enhancing retrieval precision beyond raw embedding similarity, under both original and preprocessed input settings. Thirdly, this study analyzes the contribution of domain-specific NER techniques as a preprocessing strategy, evaluating various enhancement configurations and their impact on aligning user descriptions with fault rule texts. Finally, this study investigates how different types of natural language input (e.g., precise, vague, and lengthy descriptions) affect model performance, providing insights into the system's robustness to linguistic variability.

These four dimensions collectively offer a detailed understanding of the behavior, strengths, and limitations of each method component within the defined application context.

4.5.1 Retrieval versus Classification Methods

Classification-based methods achieve the highest accuracy and robustness, making them optimal in this fixed-category, structurally homogeneous text task. As shown in Exp5, the classification approach significantly outperforms all other setups across every metric, achieving a Top1 accuracy of 0.9939 and an F1 score of 0.9940. This result aligns well with the characteristics of the task (low variability in structure and category), where classification models can fully leverage labeled training data. Though its average inference time is higher than retrieval-only methods, this overhead is often acceptable in applications where correctness is prioritized.

Retrieval-based methods provide fast inference but suffer from notable drops in accuracy. Exp1 and Exp3 highlight the efficiency of retrieval-only pipelines, with inference times of just 0.39 ms and 0.38 ms respectively. However, these methods show a clear performance deficit, with Top1 accuracies below 0.53 and F1 scores around 0.52 and 0.48. In structurally homogeneous settings, where lexical overlap may be limited, simple similarity matching often fails to capture nuanced query-class relationships, especially in the absence of contextual reranking.

Adding reranking substantially improves accuracy at the cost of increased latency heavily. Exp2 and Exp4 demonstrate that integrating a reranking component boosts performance significantly compared to naive retrieval. For example, Exp2 reaches an F1 score of 0.7358, narrowing the gap with the classification model. However, this comes with much higher average inference times, 313.21 ms and 400.69 ms respectively.

In summary, the choice of method should be guided by the specific requirements of the deployment scenario, particularly the trade-offs between accuracy, latency, and computational resources.

While the classification-based approach demonstrates the highest performance across all evaluation metrics, its relatively higher inference time may be a limiting factor in certain applications. Retrieval-based methods, especially those without reranking, offer substantial speed advantages but at the cost of reduced accuracy. The addition of reranking helps narrow this gap, albeit with increased computational overhead. Therefore, in low-resource, fixed-category, and structurally homogeneous text scenarios, classification methods are highly effective when accuracy is the primary concern, whereas retrieval-based pipelines may still be suitable under tighter latency or resource constraints, provided that some compromise in performance is acceptable.

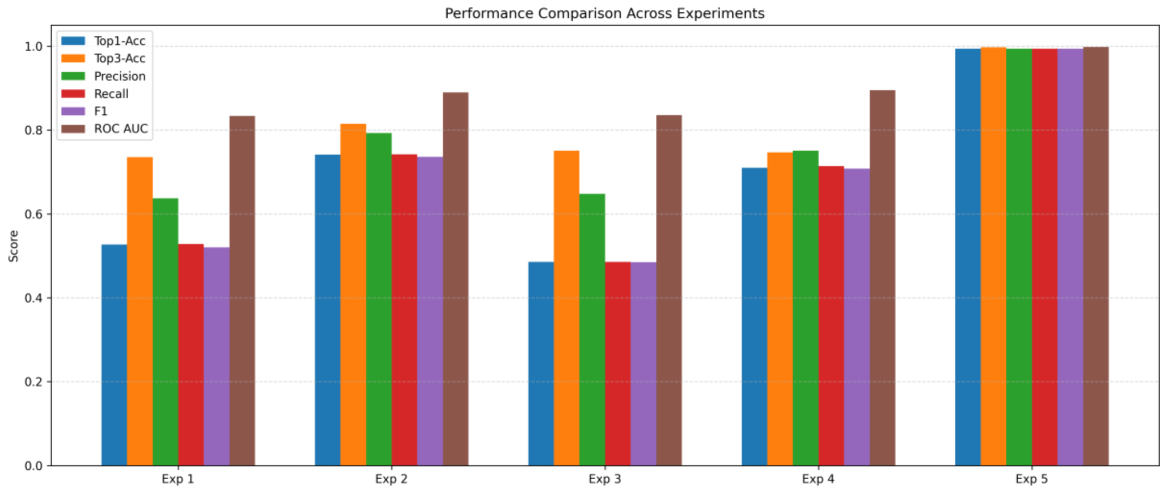

Figure 4: Performance comparison across different experimental setups.

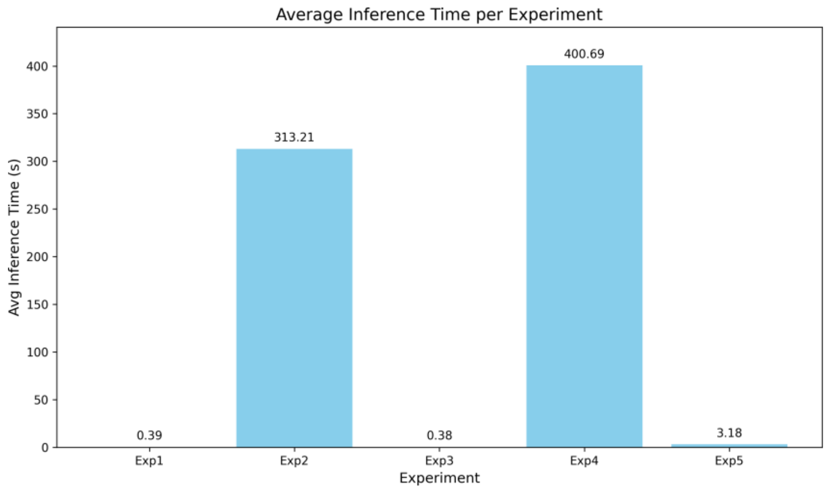

Figure 5: Average inference time across different experiments.

4.5.2 Effectiveness of Reranking

In low-resource, fixed-category, and structurally homogeneous text scenarios, the initial retrieval results are often limited by coarse-grained lexical overlaps and shallow representations. This is especially problematic in multi-class settings where categories may exhibit subtle semantic differences despite similar surface forms. Semantic reranking addresses this limitation by leveraging deeper contextual representations to refine the top candidates.

Experimental results confirm the effectiveness of this approach. Comparing Exp1 (baseline retrieval) with Exp2 (with reranker), this study observe a substantial improvement in Top-1 Accuracy (from 0.5269 to 0.7413) and F1-score (from 0.5204 to 0.7358). Precision increases from 0.6373 to 0.7931, and Macro-AUC from 0.8334 to 0.8896. Similar improvements are seen in Exp3 versus Exp4, where reranking enhances Top-1 Accuracy by over 22% and F1 by the same margin. These consistent gains suggest that reranking can significantly compensate for the limited expressive variance in low-resource texts by reordering semantically close candidates that were misranked by the initial retrieval step.

Also, these benefits extend to both raw and NER-enhanced input, indicating that reranking is robust across input representations. It plays a particularly important role in fixed-category tasks, where class labels are predefined and fine-grained distinction is crucial. Conditions under which initial retrieval often fails due to over-reliance on keyword-level similarity.

However, the cost of this semantic refinement is considerable. Inference time increases dramatically from under one ms (Exp1/Exp3) to over 300 ms (Exp2/Exp4). This computational burden is primarily due to pairwise reranking over candidate sets using large embedding models.

In summary, while semantic reranking offers clear accuracy benefits in structured and low-resource classification contexts, its utility must be weighed against inference cost. For latency-sensitive scenarios, hybrid solutions, such as using reranking only in ambiguous cases, leveraging lighter-weight rerankers, or applying approximate nearest neighbor reranking, may help balance performance and efficiency.

4.5.3 Effectiveness of NER-Based Preprocessing

In low-resource, fixed-category, and structurally homogeneous text scenarios, semantic retrieval is often challenged by variability in user input quality, ranging from overly verbose to extremely vague descriptions. Domain-specific NER offers a structured preprocessing strategy that can highlight salient information and reduce noise, potentially improving downstream matching performance. To investigate the effectiveness of this approach, this study conduct a two-stage analysis: Optimal Application Strategy and Comparative Impact Analysis.

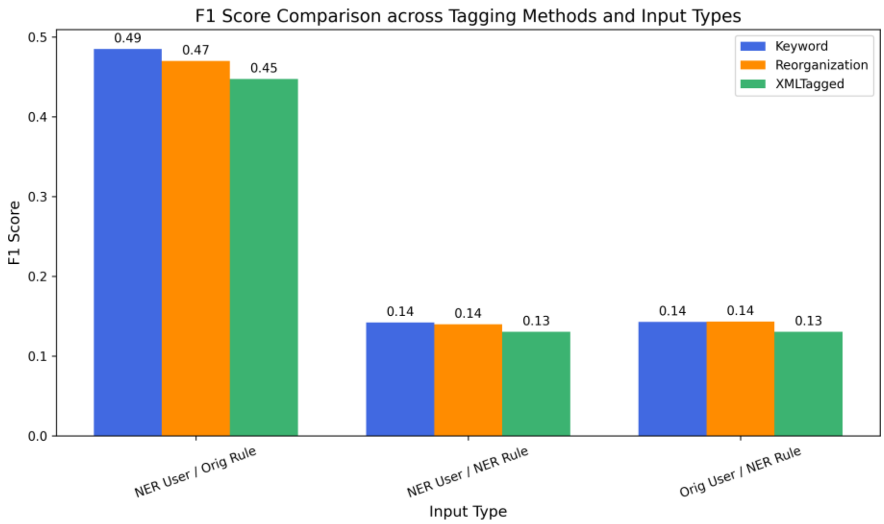

Figure 6: F1 score comparison across different tagging methods and input types.

1) Optimal Application Strategy In this section, this study interprets why keyword-based NER tagging method outperform other NER tagging methods along three dimensions: tagging strategy effectiveness, application scope sensitivity, and implications for downstream system design.

The superior performance of keyword-style augmentation can be attributed to 2 fundamental advantages that distinguish it from alternative NER integration approaches.

First, structural preservation ensures that the original syntactic and semantic coherence of user descriptions remains intact. Unlike full sentence reorganization or XML-style tagging, which introduce artificial structural modifications, keyword augmentation maintains the natural flow of language while strategically appending extracted entities. This preservation is critical because the original sentence structure often encodes implicit semantic relationships and contextual nuances that are essential for accurate similarity computation.

Second, semantic amplification without distortion allows keyword augmentation to enhance entity salience while avoiding the semantic noise introduced by more invasive rewriting methods. By treating extracted entities as supplementary signals rather than replacements, this approach creates a dual-layer representation where both the original contextual information and the highlighted entities contribute to the embedding space. This dual contribution maximizes the informativeness of the input without compromising the encoder's ability to capture holistic semantic meaning.

Applying NER asymmetrically, only to the user descriptions, proves more effective than symmetric processing of both inputs. This asymmetry likely works because user descriptions, characterized by variability, ambiguity, and informal language patterns, benefit significantly from entity-based clarification and standardization. In contrast, fault rule descriptions are typically structured, standardized, and semantically dense, requiring no additional processing. This targeted enhancement maximizes the signal-to-noise ratio by focusing computational resources where they yield the greatest marginal benefit, while avoiding unnecessary perturbations to already well-structured content.

These observations point to an important implication for retrieval-oriented NLP systems: preprocessing should be context-aware and input-specific. Rather than treating all inputs identically, designers should consider the functional role and inherent structure of each input stream. In low-resource, fixed-category tasks with asymmetric input quality, selective enhancement such as user-side NER augmentation offers a lightweight yet effective way to improve retrieval accuracy without introducing unnecessary processing complexity.

2) Comparative Impact Analysis NER with Reranker achieves a slightly higher AUC (0.8950) compared to Reranker alone (0.8896). However, despite this marginal AUC gain, the NER and Reranker approach remains substantially inferior to the standalone Reranker across all other critical metrics. This performance degradation pattern reveals that while NER augmentation provides some distributional benefits captured by AUC, it simultaneously introduces systematic errors that harm practical retrieval effectiveness.

The failure of NER-based preprocessing to consistently improve retrieval performance reveals several critical limitations that challenge conventional assumptions about entity enhancement strategies. This study analysed several reasons underlying this performance degradation (with keyword-based NER tagging method).

Domain adaptation mismatch emerges from the fundamental gap between NER models' training domains and the specific characteristics of industrial fault data. Pre-trained NER systems, typically trained on general or scientific corpora, demonstrate suboptimal recognition accuracy when applied to domain-specific industrial terminology, equipment names, and fault descriptions. This mismatch results in systematic entity misidentification and boundary errors, where the preprocessing step introduces more noise than signal, ultimately degrading rather than enhancing the quality of semantic representations.

Information dilution represents the most counterintuitive finding: in low-resource, fixed-category scenarios, the original text structure already contains sufficient discriminative information for effective retrieval. The addition of NER-processed entities does not introduce new discriminative signals but rather dilutes the concentration of existing meaningful patterns. In embedding spaces with limited dimensionality, this dilution effect reduces the model's ability to distinguish between semantically similar but categorically different fault types, leading to decreased retrieval precision.

Furthermore, retrieval signal interference occurs when NER augmentation creates false similarity signals between unrelated fault cases that happen to share extracted entities. In industrial contexts, superficial entity overlap (such as common equipment names or general fault symptoms) may mask deeper semantic differences that are crucial for accurate fault diagnosis. This interference is particularly problematic in embedding-based retrieval systems that rely heavily on lexical and semantic overlap patterns.

These findings establish that in constrained, domain-specific scenarios with inherently structured data, NER enhancement operates beyond its effectiveness boundary. The marginal complexity introduced by entity processing exceeds its marginal utility, resulting in a net negative impact on retrieval performance. This challenges the widespread assumption that entity extraction universally improves semantic understanding, highlighting the need for context-aware preprocessing strategies that consider domain characteristics, data structure, and task requirements.

4.5.4 Analysis of Negative Interactions Among System Components

The experimental results reveal an unexpected phenomenon where the combination of NER enhancement and reranking (Exp4) yields inferior performance compared to reranking alone (Exp2). This counterintuitive finding warrants careful analysis as it challenges conventional assumptions about component synergy in information retrieval systems.

Specifically, the comparison between Exp2 and Exp4 demonstrates a clear case of negative interaction between system components. While Exp2 achieves a Top-1 accuracy of 0.7413, the addition of NER enhancement in Exp4 results in a degraded performance of 0.7097, representing a 4.3% decrease in accuracy. This degradation occurs despite the theoretical complementarity of the two enhancement strategies, suggesting systematic interference between components rather than additive benefits.

Two primary mechanisms may explain the observed negative interactions.

First of all, cumulative error propagation occurs through the sequential processing pipeline. Errors introduced during NER processing—including entity misclassification or boundary detection failures—propagate to the retrieval stage and become amplified during reranking. This cascading effect explains why the combined system underperforms either component in isolation.

Then, semantic representation inconsistencies emerge from the conflicting assumptions underlying different components. NER uses the SciBERT model, while the embedding representation relies on the BGE model. Moreover, the NER module prioritizes entity-centric representations, while the reranker employs broader contextual semantic understanding. These incompatible frameworks create internal contradictions that degrade overall system coherence.

These findings have significant implications for system design in information retrieval. The documented negative interactions suggest that component integration requires careful consideration of semantic compatibility rather than simple concatenation of high-performing modules. The discovery of systematic negative interactions represents a novel contribution to understanding multi-component system behavior and highlights the need for component compatibility assessment in system design.

4.5.5 Effectiveness of Input User Description Type

This section discusses the implications of varying user input types on model performance and system design.

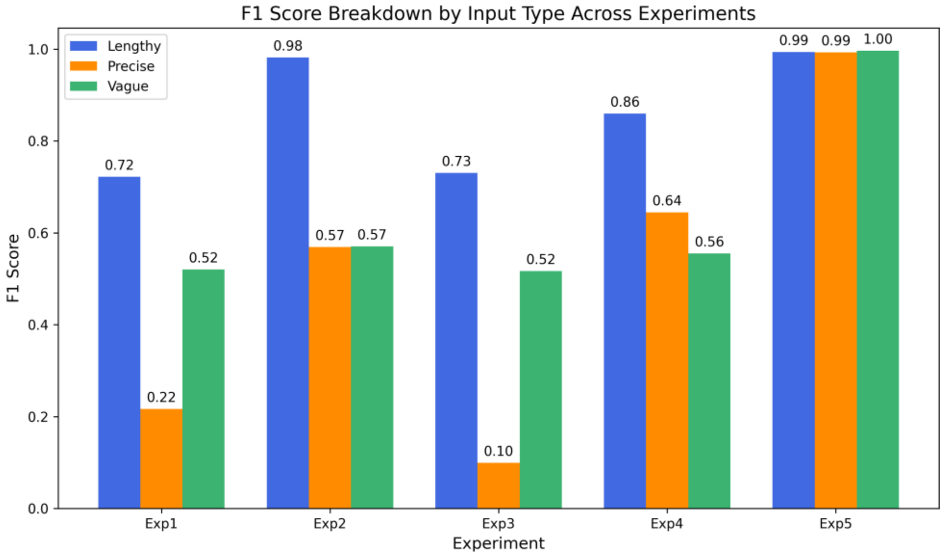

Figure 7: F1 score breakdown by input type across different experiments.

A somewhat counterintuitive phenomenon arises in the semantic matching task: one would expect that more precise machine-generated descriptions (more professional and technology-oriented) would yield better retrieval and matching performance. However, empirical observations indicate the opposite: lengthy, elaborative expressions demonstrate superior matching accuracy compared to their precise counterparts. To illustrate this, consider the following example:

Table 13: Class Label and Corresponding Descriptions for Internal Software Error in Industrial Dehumidifiers

Category	Content
Class Label	*"The unit has stopped. Internal software error related text is shown on the Alarms menu.*
Lengthy user description (correctly matched)	Example 1: "I checked the dehumidifier after a reported fault and saw a blinking red light. The display was stuck showing Internal error. We might need to contact support to diagnose the issue." Example 2: "Everything was normal until mid-morning. Suddenly the system stopped, and the red lamp came on. The error message reads 'Internal software error'. We tried restarting it, but the error comes back immediately."
Precise user description (wrongly matched)	Example 1: "The industrial dehumidifier (DRYMAX-300) has stopped functioning. The red alarm lamp on the control panel is blinking. The alarm screen displays Internal software error. Current room humidity is 85%, temperature is 26°C. Voltage within normal range. Likely cause: firmware malfunction, corrupted control logic, or memory overflow error." Example 2: "Functionality has ceased in the AirPro-DH800 industrial dehumidifier. The red alarm lamp on the control panel blinks, and the alarm screen shows software error. Current environmental conditions: 85% humidity and 24°C temperature. The main power supply is stable. Potential causes include firmware errors, corrupted control logic, or memory overflow issues."

The observed phenomenon, where lengthy user descriptions outperform precise ones in semantic matching tasks, can be explained through several complementary theoretical frameworks.

The primary mechanism underlying this phenomenon is semantic over-specification, where excessive technical precision creates semantic distance from standardized class labels. Industrial fault classification systems deliberately employ generalized, template-based descriptions designed for broad applicability across equipment variants and operational contexts. Class labels such as "Internal software error" are intentionally simplified to encompass various manifestations of software-related failures. When user inputs contain highly specific technical jargon (e.g., "DRYMAX-300," "memory overflow error code 0x4B2F"), the embedding space representation diverges significantly from these standardized templates. The additional precision introduces semantic noise that interferes with core concept matching. While terms like "corrupted control logic" and "memory overflow error" are technically accurate, they represent domain-specific vocabulary that diverges from the simpler language employed in class labels, creating what we term "lexical interference."

Furthermore, the superior performance of verbose descriptions can be attributed to semantic pathway redundancy, where multiple expressions of the same underlying concept create resilient matching conditions. The verbose example demonstrates this principle by repeatedly referencing the operational failure through varied linguistic expressions (”stopped working,” ”completely shut down,” ”can’t get it to restart”), visual indicators (”red light,” ”blinking constantly”), and error manifestations (”error message,” ”internal software error”). This redundancy functions as a semantic error correction mechanism within the transformer-based language models, providing multiple independent channels for successful alignment with target labels. Modern models like BGE derive semantic understanding through contextual embeddings, making them particularly sensitive to the narrative flow and contextual scaffolding present in natural language expressions. The conversational progression from problem observation to symptom identification to attempted resolution creates a coherent semantic narrative that facilitates multiple matching pathways. In contrast, precise descriptions create brittle matching conditions where slight semantic misalignment can result in complete matching failure due to their reliance on concentrated information channels. The concentrated nature of technical terminology creates semantic bottlenecks where matching success depends critically on exact terminological alignment, whereas verbose descriptions distribute conceptual information across multiple linguistic expressions, increasing the probability of successful matching.

These findings have significant implications for designing semantic matching systems in industrial domains. System architectures should accommodate both formal technical language and informal conversational patterns. The observed preference for conversational patterns may also reflect the training data composition of underlying language models, highlighting the importance of balanced representation between formal and informal technical language in model development and fine-tuning strategies.

4.5.6 Per-class Performance Analysis

The disparity in performance across classes can be partly attributed to the clarity and specificity of the issue descriptions associated with each class. As this section involves proprietary diagnostic rules, the full descriptions of Rule 1 to Rule 17 are not disclosed; however, a few examples are provided below to aid understanding.

A closer examination of per-class performance reveals substantial variation, which can be partly attributed to the linguistic and structural properties of the diagnostic rule descriptions. Through a analysis and observation of the rule corpus (Rule 1 to Rule 17, content not fully disclosed for confidentiality), this study observes two characteristics that influence model:

- Lexical redundancy with keyword-level variance: Some rule descriptions exhibit highly repetitive phrasing, differing primarily at the level of one or two discriminative keywords (e.g., sensor type or fault location).
- Feature overlap across rules: Some rules describe composite symptoms, resulting in shared features across multiple classes. This overlap reflects imperfections in data labeling or rule design, and may also stem from limitations in annotation granularity or the quality of the underlying data. Nevertheless, it remains an inherent characteristic of the current dataset.

Considering the above data characteristics, and based on the performance shown in Table 12, it can categorize the performance of each class into three tiers.

1) Low-performing classes(e.g., Class 1) Low-performing categories are often associated with vague or multi-causal descriptions. For example, Class 1 (“The unit has stopped. The display window and indicators are not lit.”) exhibits poor performance due to its general phrasing and substantial semantic overlap with other classes. Consider Class 2 as an example (“The unit has stopped. The unit is in AUTO mode. Operation mode shows STANDBY.”). The difficulty arises because both classes reference a system stoppage, and the visual symptoms described in Class 1 (i.e., an unlit display and inactive indicators) can easily be misinterpreted as signs of the standby mode described in Class 2. The absence of precise technical terminology or distinctive error cues in Class 1 further contributes to the model’s inability to clearly differentiate it from other classes.

2) High-performing classes (e.g., 12) In contrast, high-performing categories correspond to rule descriptions that are both lexically distinct and semantically unambiguous. For example, Class 12 (“The unit has stopped.’Sensor Fail’ similar text is shown on the ’Alarms menu’.”) achieves high accuracy because it contains distinctive technical termi- nology (”Sensor Fail”) and unambiguous operational states that rarely appear in other categories.

3) Intermediate and improving classes (e.g., Class 6, 7) Classes such as 6 and 7 demonstrate intermediate performance. These classes include alarm messages like “Sensor fault Temperature before heater” and “Sensor fault Temperature Sensor.” While such messages share a repeated sentence scaffold (“The unit has stopped... alarm lamp blinking... text is shown on the Alarms menu”), they differ in the fault-specific terms. The model’s ability to distinguish between them improves with increased training or exposure to richer representations, though the shared structure and overlapping semantics remain challenging.

5 Conclusion and Future Work

In this study, this study conducted a systematic comparison between retrieval-based and classification-based approaches for mapping user descriptions to domain-specific rule labels. Our results highlight clear performance trade-offs across different experimental setups.

Based on our systematic comparative analysis, classification-based methods demonstrate superior performance over retrieval-based approaches in the specific deployment conditions examined. This study conducted a comprehensive comparison between retrieval-based and classification-based approaches for mapping user descriptions to domain-specific rule labels, revealing clear performance trade-offs across different experimental setups.

The empirical evidence consistently shows that classification-based models outperformed retrieval-based pipelines across all evaluation metrics when operating under realistic constraints of limited resources, fixed categories, and structurally homogeneous text. These results validate the strength of end-to-end supervised learning when task-specific training data is available, particularly in scenarios where subtle semantic distinctions must be captured within a constrained categorical framework.

Secondly, domain-specific NER preprocessing demonstrates limited effectiveness and may introduce performance degradation. NER-based enhancement fails to improve retrieval performance consistently due to domain adaptation mismatch, information dilution, and retrieval signal interference. This establishes effectiveness boundaries for preprocessing techniques in constrained scenarios.

Thirdly, negative interactions between system components can occur despite theoretical complementarity. The combination of NER and reranking yields inferior performance compared to reranking alone. This demonstrates that component integration requires semantic compatibility rather than simple concatenation of high-performing modules.

Finally, the analysis reveals that lengthy, conversational user descriptions outperform precise, technical specifications, providing new insights into semantic matching mechanisms. Two theoretical frameworks explain this phenomenon: semantic over-specification, where excessive technical precision creates semantic distance from standardized labels, and semantic pathway redundancy, where multiple expressions create resilient matching conditions.

5.1 Contributions

This study presents the following main contributions to the field of semantic matching in constrained industrial environments:

- **Systematic comparison under constrained industrial scenarios**: This research is the first to conduct a comprehensive comparison between classification-based and retrieval-based semantic matching methods under the combined constraints of low-resource availability, structural homogeneity, and fixed-category taxonomies. Unlike previous studies that focus on resource-rich or narrowly defined domains, this work addresses a practically significant yet underexplored setting and provides methodological guidance for handling patterned industrial text.

- **Identification of the limitations of NER-based enhancement**: Through systematic experiments, this study finds that NER does not universally enhance performance and, in certain configurations, leads to performance degradation. The underlying mechanisms are examined, including excessive entity density that disrupts semantic coherence, semantic misalignment due to domain shifts between pre-trained models and industrial texts, and structural perturbations caused by entity rewriting. These findings establish a theoretical foundation for understanding the conditional effectiveness of NER, offering a correction to the commonly held assumption that entity enrichment is always beneficial.

- **Discovery of counterintuitive effects related to input precision**: The study reveals that precise user inputs do not always result in better matching performance; in some configurations, vague inputs perform more effectively. This challenges the conventional belief that increased input specificity improves semantic alignment. The observed phenomenon is explained by the tendency of precise inputs to introduce irrelevant details that hinder generalization, and by the semantic mismatch between detailed queries and generalized rule expressions.
- **Analysis of negative interactions among system components**: Experimental results demonstrate that combining NER with reranking modules can, in some cases, reduce system performance compared to using either component independently. This study attributes the effect to accumulation of errors across processing stages, and inconsistent interpretations of input semantics across modules. These insights contribute to a deeper understanding of interaction effects in modular NLP system design.
- **Development of a deployment-oriented decision framework**: Based on the empirical findings, this research proposes a decision framework specifically designed for semantic matching in industrial applications characterized by domain specificity, structural repetition, and limited data resources. This framework supports practitioners in selecting appropriate strategies for system architecture, thereby reducing deployment risks and development costs.

5.2 Limitations

Several limitations of this study must be acknowledged that may affect the interpretation and generalizability of the findings:

- **Data Quality and Authenticity:** A portion of the dataset used in this study may have been synthesized or augmented using language models, which introduces the possibility of distributional bias that could favor transformer-based encoders like BERT. The data augmentation process may have inadvertently created variations of original samples that appeared in both training and testing sets, potentially compromising test set independence and leading to overly optimistic evaluation results.
- **Dataset Representativeness:** The evaluation datasets, while representative of the targeted scenarios, may not fully capture the linguistic diversity and complexity variations present in real-world industrial environments. The extent to which findings generalize to broader industrial fault diagnosis contexts requires further validation.

- **Experimental Independence:** The experimental design includes instances where certain experiments rely on optimal parameter combinations identified in prior experiments. This sequential dependency reduces the independence between experimental results and may affect the fairness of cross-paradigm comparisons, potentially inflating perceived improvements in subsequent experiments.
- **Evaluation Scope:** While the study focuses on standard performance metrics (accuracy, precision, recall, F1-score), it does not comprehensively address specialized industrial evaluation criteria such as cost-sensitive performance measures, real-world deployment constraints, or user experience factors that may be crucial in practical applications.
- **Temporal Validation:** The study assumes static performance characteristics and does not account for potential performance degradation over time due to concept drift, evolving terminology, or changing user behavior patterns in industrial settings.

5.3 Future Work

Future research will explore several promising directions to extend the findings of this study:

- **Hybrid Architectures:** Future work will investigate hybrid retrieval and classification pipelines, where retrieval is used to narrow down candidate outputs, followed by lightweight classification or reranking. Such architectures may improve the trade-off between efficiency and accuracy, particularly under limited computational resources or real-time constraints.
- **Input Variability Modeling:** A more detailed analysis of user input variability, including domain knowledge, linguistic register, and intent formulation, may reveal insights that inform the design of more robust models and effective input preconditioning strategies.
- **Dataset Scope and Generalization:** The current study is limited to a single industrial domain with 17 predefined fault categories and constrained structural characteristics. Future research should evaluate the proposed methods across multiple domains, datasets of varying sizes, and fault scenarios of differing complexity to better understand the boundaries of generalizability.
- **Cross-Domain and Multilingual Evaluation:** To assess deployment readiness in practical settings, future studies should expand evaluation to include multilingual and cross-domain datasets, which are representative of real-world variability.
- **Component Interaction Frameworks:** Although negative interactions between components such as named entity recognition and reranking were observed, their underlying mechanisms require further investigation. Future research should develop formal analytical frameworks for identifying, predicting, and mitigating such component-level interference in multi-stage information retrieval systems.

Bibliography

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. “SciBERT: A Pretrained Language Model for Scientific Text”. In: *EMNLP*. Association for Computational Linguistics, 2019. url: https://www.aclweb.org/anthology/D19-1371.

[2] Huiyao Chen et al. “Retrieval-style in-context learning for few-shot hierarchical text classification”. In: *Transactions of the Association for Computational Linguistics* 12 (2024), pp. 1214–1231.

[3] Jianlv Chen et al. *BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation*. 2024. arXiv: 2402.03216 [cs.CL].

[4] Jianlv Chen et al. *BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation*. 2024. arXiv: 2402.03216 [cs.CL].

[5] Yahui Chen. “Convolutional neural network for sentence classification”. MA thesis. University of Waterloo, 2015.

[6] Aaron M Cohen, Jianji Yang, and William R Hersh. “A Comparison of Techniques for Classification and Ad Hoc Retrieval of Biomedical Documents.” In: *TREC*. 2005.

[7] Brian De Silva et al. “Semantic matching for text classification with complex class descriptions”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023, pp. 7654–7680.

[8] Xingyu Deng, Rong Luo, and Junshan Li. “Similarity matching algorithm of equipment fault diagnosis based on CBR”. In: *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE. 2015, pp. 998–1002.

[9] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 2019, pp. 4171–4186.

[10] Chunning Du et al. “Syntax-type-aware graph convolutional networks for natural language understanding”. In: *Applied Soft Computing* 102 (2021), p. 107080.

[11] Henning Holle and Robert Rein. “EasyDIAg: A tool for easy determination of interrater agreement”. In: *Behavior research methods* 47 (2015), pp. 837–847.

[12] Jie Hu et al. “Text semantic matching algorithm based on the introduction of external knowledge under contrastive learning”. In: *International Journal of Machine Learning and Cybernetics* 16.1 (2025), pp. 741–753.

[13] Shayekh Bin Islam et al. “Open-rag: Enhanced retrieval-augmented reasoning with open-source large language models”. In: *arXiv preprint arXiv:2410.01782* (2024).

[14] Patrick Lewis et al. “Retrieval-augmented generation for knowledge-intensive nlp tasks”. In: *Advances in neural information processing systems* 33 (2020), pp. 9459–9474.

[15] Chaofan Li et al. *Making Large Language Models A Better Foundation For Dense Retrieval*. 2023. arXiv: 2312.15503 [cs.CL].

[16] Chaofan Li et al. *Making Text Embedders Few-Shot Learners*. 2024. arXiv: 2409.15700 [cs.IR]. url: https://arxiv.org/abs/2409.15700.

[17] Hang Li, Jun Xu, et al. “Semantic matching in search”. In: *Foundations and Trends® in Information Retrieval* 7.5 (2014), pp. 343–469.

[18] Jiaxin Liu, Yi Yang, and Kar Yan Tam. “Beyond surface similarity: Detecting subtle semantic shifts in financial narratives”. In: *arXiv preprint arXiv:2403.14341* (2024).

[19] Yinhan Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).

[20] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[21] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[22] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[23] Xipeng Qiu et al. “Pre-trained models for natural language processing: A survey”. In: *Science China technological sciences* 63.10 (2020), pp. 1872–1897.

[24] Nils Reimers and Iryna Gurevych. “Sentence-bert: Sentence embeddings using siamese bert-networks”. In: *arXiv preprint arXiv:1908.10084* (2019).

[25] Stephen Robertson, Hugo Zaragoza, et al. “The probabilistic relevance framework: BM25 and beyond”. In: *Foundations and Trends® in Information Retrieval* 3.4 (2009), pp. 333–389.

[26] Fatemeh Sarvi et al. “A comparison of supervised learning to match methods for product search”. In: *arXiv preprint arXiv:2007.10296* (2020).

[27] Darya Shlyk et al. “REAL: A Retrieval-Augmented Entity Linking Approach for Biomedical Concept Recognition”. In: *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*. Association for Computational Linguistics. 2024, pp. 380–389.

[28] Prashant Upadhyay et al. “A comprehensive survey on answer generation methods using NLP”. In: *Natural Language Processing Journal* 8 (2024), p. 100088.

[29] Nikil Wale, Ian A Watson, and George Karypis. “Comparison of descriptor spaces for chemical compound retrieval and classification”. In: *Knowledge and Information Systems* 14 (2008), pp. 347–375.

[30] Dongsheng Wang et al. “PDAM-FAQ: Paraphrasing-based Data Augmentation and Mixed-Feature Semantic Matching for Low-Resource FAQs”. In: *IEEE Access* (2024).

[31] Bella Hass Weinberg. “Library classification and information retrieval thesauri: comparison and contrast”. In: *Cataloging & Classification Quarterly* 19.3-4 (1995), pp. 23–44.

[32] Shitao Xiao et al. *C-Pack: Packaged Resources To Advance General Chinese Embedding*. 2023. arXiv: 2309.07597 [cs.CL].

[33] Lee Xiong et al. “Approximate nearest neighbor negative contrastive learning for dense text retrieval”. In: *arXiv preprint arXiv:2007.00808* (2020).

[34] Chunlin Xu et al. “Treelstm with tag-aware hypernetwork for sentence representation”. In: *Neurocomputing* 434 (2021), pp. 11–20.

[35] Dian Yu et al. “Few-shot intent classification and slot filling with retrieved examples”. In: *arXiv preprint arXiv:2104.05763* (2021).

[36] Jianxing Yu et al. “Multi-hop reasoning question generation and its application”. In: *IEEE Transactions on Knowledge and Data Engineering* 35.1 (2021), pp. 725–740.

[37] Junru Zhou, Zuchao Li, and Hai Zhao. “Parsing all: Syntax and semantics, dependencies and spans”. In: *arXiv preprint arXiv:1908.11522* (2019).

Appendix

This is the prompt feed to chatgpt for generating the user description data:

Prompt

You are an AI language model tasked with simulating realistic failure descriptions written by industrial maintenance personnel or factory technicians. Users describe equipment problems in different ways—some are technical and concise, others vague or overly detailed.
1. Task
Given a base failure rule, generate 500 unique and natural-sounding failure descriptions, covering a range of expression styles.
2. Failure Rule
`{This is a placeholder. The rule is internal and not disclosed here.}`
3. Style Categories
Descriptions must be evenly distributed across the following three styles:
1) Precise descriptions: Technical, clear, and concise. May include model numbers, error codes, sensor values, and suspected causes.
2) Vague descriptions: Non-technical and unclear. May only describe symptoms or express confusion, with little detail.
3) Lengthy descriptions: Overly detailed or anecdotal. May include irrelevant background, speculation, or user opinions.
4. Requirements
- Ensure a balanced number of descriptions per style.
- Vary sentence length, tone, and perspective (e.g., operator, technician).
- Some descriptions should include numerical data or error codes; others should not.
- Avoid duplication in wording or phrasing.
- Each description should clearly reflect the base failure but be phrased in a unique and human-like way.

A total of 8,500 failure descriptions were generated by applying the prompt across 17 different failure rules (500 descriptions per rule). After deduplication, the resulting dataset comprised 8,195 unique entries.