

Article

Enterprise E-Mail Classification Using Instruction-Following Large Language Models

Ahmet Çağrı Sarıyıldız ¹  and Şafak Durukan-Odabaşı ^{2,*} ¹ Department of Cybersecurity, Bahçeşehir University, Istanbul 34349, Türkiye; acsariyildiz@gmail.com² Department of Computer Engineering, Istanbul University-Cerrahpaşa, Istanbul 34320, Türkiye

* Correspondence: safak.odabasi@iuc.edu.tr

Abstract

Enterprise e-mail corpora contain heterogeneous and domain-specific content that poses challenges for conventional supervised Natural Language Processing (NLP) approaches due to class imbalance, evolving terminology, and limited labeled data. This study examines the use of instruction-following Large Language Models (LLMs) for enterprise e-mail classification under realistic operational conditions. The study evaluates instruction-based classification and semantic enrichment derived from distributional similarity as two complementary approaches for distinguishing technical from nontechnical messages. The approaches are assessed on a large-scale enterprise e-mail corpus and validated using a manually annotated subset. The results indicate that instruction-following LLMs provide stable contextual reasoning across diverse message structures, while semantic enrichment improves coverage of previously unseen technical expressions. Overall, the study presents an applied NLP framework for enterprise e-mail classification, with attention to interpretability, scalability, and robustness in real-world organizational settings.

Keywords: natural language processing; enterprise e-mail classification; large language models; instruction-following models; applied machine learning

1. Introduction

Recent progress in Large Language Models (LLMs) has significantly enhanced the performance of Natural Language Processing (NLP) systems across a wide range of standard tasks, including text classification and named entity recognition. With the growing accessibility of advanced models, such as those offered by for-profit organizations like OpenAI and DeepSeek through Application Programming Interface (API) services, developers and organizations can now integrate cutting-edge NLP functionalities with relative ease. However, leveraging these general-purpose models within enterprise settings introduces a distinct set of challenges that extend beyond mere model accuracy. One of the primary concerns is the dependency on proprietary API services. Commercial models often entail considerable costs and introduce operational dependencies on external platforms, which may not align with the strategic or regulatory priorities of many organizations. Moreover, such APIs frequently offer limited customization options, restricting their adaptability to highly specific domains. In enterprise contexts where domain relevance and interpretability are essential, this lack of flexibility can diminish the practical utility of otherwise powerful models.

Another significant limitation is the scarcity of high-quality, labeled datasets within specialized domains. While general-purpose LLMs are trained on vast corpora covering



Academic Editor: Alberto Gil Solla,
Yolanda Blanco Fernández, José Carlos
López Ardao

Received: 6 January 2026

Revised: 14 February 2026

Accepted: 20 February 2026

Published: 24 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

diverse topics, they are rarely optimized for the nuanced language patterns and terminology used in specific organizational contexts. Data privacy constraints further widen this gap; many institutions manage sensitive or proprietary information that cannot be externally shared, thereby limiting the ability to fine-tune or evaluate models on in-domain data.

The broader research landscape reflects these limitations. Although LLMs are well-represented in the literature for general NLP benchmarks, their application in domain-specific scenarios, particularly in real-world corporate environments, remains relatively underexplored. This imbalance necessitates empirical studies that examine both the opportunities and constraints of utilizing LLMs in specialized settings.

In this study, we respond to this need by examining the Enron e-mail dataset [1,2], a publicly available and widely studied corpus of corporate communication, as a testbed for exploring domain-specific NLP applications. Although the dataset is a legacy corpus, it remains a widely recognized and realistic benchmark for enterprise communication, providing a valuable resource for evaluating NLP techniques tailored to organizational contexts.

Our focus lies in identifying e-mails that are functionally relevant to an organization's technical infrastructure within internal communication. This type of insight is crucial for tasks such as Information Technology (IT) auditing, infrastructure planning, and system monitoring. To this end, we investigate two distinct methodological approaches: one utilizing vector-based keyword expansion with Sense2Vec [3], and the other employing the Fine-Tuned Language Net based on the Text-to-Text Transfer Transformer (FLAN-T5) model [4], a transformer-based encoder–decoder architecture optimized for instruction-following tasks. Through both qualitative and quantitative assessments, we aim to evaluate the strengths and limitations of each approach and, in doing so, contribute to the development of more effective NLP strategies for enterprise applications.

In enterprise contexts, the distinction between technical and nontechnical communication is more than an academic exercise; it has clear practical implications. Recent studies have demonstrated the relevance of NLP-driven e-mail classification for risk scoring, spam detection, and handling data imbalance [5–7]. Building on these advances, our study focuses on technical communication within enterprise e-mail corpora, a task that directly supports IT auditing, infrastructure monitoring, and compliance workflows. For example, identifying messages that reference system versions, configuration details, or dependencies can enrich asset management systems and knowledge bases, while also improving visibility for cybersecurity monitoring. By framing the problem as a domain-specific NLP task, this paper not only analyzes two complementary approaches, namely FLAN-T5 and Sense2Vec, but also provides a transparent and reproducible baseline for future enterprise-oriented NLP research, and adopts a task-oriented perspective and focuses on practical trade-offs between contextual reasoning and semantic coverage in enterprise NLP applications.

In this study, a technical e-mail is defined as a message containing infrastructure-relevant entities (e.g., servers, databases, network components, software tools) in conjunction with operational or configuration-related actions that indicate functional relevance to enterprise IT systems.

The main contributions of this study can be summarized as follows:

- A reproducible, enterprise-oriented evaluation framework is presented in which instruction-following classification and semantic expansion are implemented independently and assessed under identical operational conditions.
- A comparative analysis of instruction-following large language model-based classification and semantic keyword expansion is conducted under realistic operational constraints.
- Practical trade-offs between precision, recall, and interpretability relevant to real-world enterprise deployment scenarios are discussed.

The structure of this paper is as follows: In Section 2, studies on e-mail classification, keyword extraction, and entity recognition are reviewed. Section 3 defines the problem of identifying technical content in large-scale e-mail data. Section 4 explains the preprocessing steps applied to the Enron dataset. Section 5 presents the use of FLAN-T5-Large for classifying e-mails. In Section 6, the Sense2Vec-based keyword extraction and classification method is introduced. Section 7 provides the experimental results and comparative evaluation of the FLAN-T5- and Sense2Vec-based models, including performance on the annotated dataset. Finally, Section 8 discusses the implications of the findings, highlights limitations, and outlines potential directions for future research.

2. Related Work

Research on e-mail analysis has evolved considerably, progressing from early spam and phishing filters based on classical machine learning to more recent applications of deep learning and LLMs. While these advances have strengthened defenses against unsolicited communication, the task of analyzing enterprise e-mail corpora remains comparatively underexplored. This gap is not only conceptual but also operationally significant, since extracting domain-relevant information can directly support practical functions such as IT auditing, infrastructure monitoring, and compliance management.

Early contributions established the foundation for automated spam filtering and anomaly detection. Chandola et al. [8] provided one of the earliest surveys on anomaly detection, while Buczak and Guven [9] reviewed machine learning methods for cybersecurity applications, including spam detection. Within e-mail classification specifically, early models such as Naïve Bayes, decision trees, and support vector machines achieved reasonable accuracy but were constrained by their reliance on handcrafted features and limited adaptability to evolving adversarial strategies.

With the emergence of deep learning, more advanced architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) were applied to spam and phishing detection, improving feature representation through automatic extraction from raw text. Zhang et al. [10] demonstrated CNN-based models for spam classification, while Alshingiti et al. [11] developed CNN-, Long Short-Term Memory (LSTM)-, and hybrid LSTM-CNN-based approaches for phishing detection, reporting competitive accuracy across architectures. Atawneh and Aljehani [12] further showed that Bidirectional Encoder Representations from Transformers (BERT)- and LSTM-based hybrid methods can improve phishing classification performance.

More recent works have shifted toward embedding-based and transformer-driven approaches. Li et al. [13] provided a comprehensive survey of deep learning for named entity recognition (NER), framing how context-aware embeddings could support security-related tasks. Sammet et al. [14] extended this direction by presenting a BERT-based keyword extraction framework tailored for specialized domains. In parallel, Shang et al. [15] introduced AutoPhrase, an unsupervised phrase mining system, while Zhang et al. [16] proposed a context-aware phrase mining model using attention mechanisms to refine extracted domain terms.

Keyword extraction has also been advanced by semantic post-processing and embedding-based methods. Altuncu et al. [17] demonstrated that combining part-of-speech filtering with semantic similarity improves keyword quality across corpora. Khan et al. [18] analyzed contextual embedding approaches such as KeyBERT, showing that embedding-based extraction consistently outperforms statistical baselines, particularly for short and technical texts.

Yu et al. [19] also emphasized the need to systematically evaluate and improve the robustness of named entity recognition systems under domain-specific variation. Automated testing frameworks that introduce controlled perturbations and semantic substitutions

have demonstrated how entity-level models may degrade under lexical shifts common in enterprise communication. These findings reinforce the importance of semantically informed keyword expansion and context-aware extraction strategies when operating on heterogeneous corporate corpora.

Transformers and LLMs have accelerated this progress further. Labonne and Moran [20] proposed Spam-T5, adapting FLAN-T5 for few-shot spam detection. Patel et al. [21] benchmarked LLMs for phishing detection, identifying trade-offs between precision and efficiency. Altwaijry et al. [22] evaluated various deep learning architectures for phishing e-mail detection and demonstrated that an augmented 1D-CNN variant achieved a favorable balance of precision and recall under realistic conditions, while Tarapiah et al. [23] compared fine-tuned LLMs against classical Machine Learning (ML) models in enterprise-like settings and reported that LLMs offered improved recall without dramatically sacrificing efficiency.

In [24] LLM-driven e-mail classification frameworks have extended transformer-based architectures to enterprise-scale processing scenarios. Hybrid systems combining pre-trained embeddings with neural classifiers have demonstrated operational feasibility while also incorporating explanation-generation components to enhance interpretability. Although predominantly focused on spam detection, these studies illustrate how instruction-following and transformer-based models can be integrated into real-world e-mail pipelines under deployment constraints.

Beyond malicious content detection, enterprise communication corpora have also been studied. The Enron dataset [1,2] remains the most widely used benchmark for organizational e-mails. Although it has been applied to spam filtering, topic modeling, and entity recognition, relatively few works have examined its potential for extracting technical infrastructure information that directly supports enterprise tasks such as IT auditing, system monitoring, and compliance reporting.

Building on these enterprise-focused perspectives, recent research has increasingly explored production-oriented e-mail processing pipelines. Industry-driven architectures such as the Advanced Messaging Platform (AMP) [25] demonstrate how large-scale enterprise e-mail streams can be processed through integrated modules for intent recognition, entity extraction, workflow triggering, and human-in-the-loop validation. These systems highlight practical deployment considerations, including scalability, auditability, and operational robustness in real-world corporate environments.

In parallel, the integration of large language models with enterprise data infrastructures has gained attention as a means of transforming unstructured communication into structured organizational knowledge. Frameworks for LLM-powered enterprise knowledge graphs [26] have shown how entities and relationships extracted from heterogeneous sources—including e-mail corpora—can support contextual analytics, expertise discovery, and decision intelligence. This perspective situates e-mail-level classification as a foundational step within broader enterprise intelligence pipelines.

Complementary perspectives [27] have further examined the combination of large language models with enterprise knowledge graphs to enhance semantic grounding and contextual consistency. By linking generative models with symbolic organizational structures, these approaches seek to balance flexibility with interpretability in enterprise NLP systems. Although primarily conceptual, such discussions underscore the growing convergence between LLM-based extraction pipelines and structured knowledge representations.

Recent empirical studies [28] have also examined how large language models are being adopted in knowledge-intensive organizational settings. Survey-based and observational analyses indicate that LLMs are increasingly integrated into drafting, summarization, analytical reasoning, and information synthesis tasks within enterprise workflows. These findings highlight both the productivity potential and the operational limitations of deploy-

ing LLMs in professional environments. Such observations reinforce the need for reliable, domain-specific classification mechanisms when LLMs are deployed in enterprise communication contexts. Our study addresses this underexplored area by empirically analyzing two complementary approaches—Sense2Vec for keyword expansion and FLAN-T5 for instruction-following classification—within a unified enterprise NLP pipeline.

3. Problem Formulation and Research Objectives

This study addresses two primary challenges: (1) the identification of e-mails containing technically relevant content within the Enron dataset, and (2) the extraction of structured information about technical infrastructure from the identified communications. In practical terms, this means recognizing, labeling, and organizing knowledge about the IT systems, tools, and configurations mentioned in corporate correspondence. The overarching goal is to design a method capable of automatically classifying e-mails according to their technical relevance and subsequently extracting detailed information related to the systems, configurations, and tools referenced within these messages. This aligns with the broader aim of improving enterprise knowledge management and enabling automated insights for IT auditing and cybersecurity monitoring.

To illustrate the scope of the problem, consider Figure 1: an internal e-mail sent by Bob to Alice via the organization’s corporate communication system. This example typifies the type of interaction that is targeted for analysis in this study.

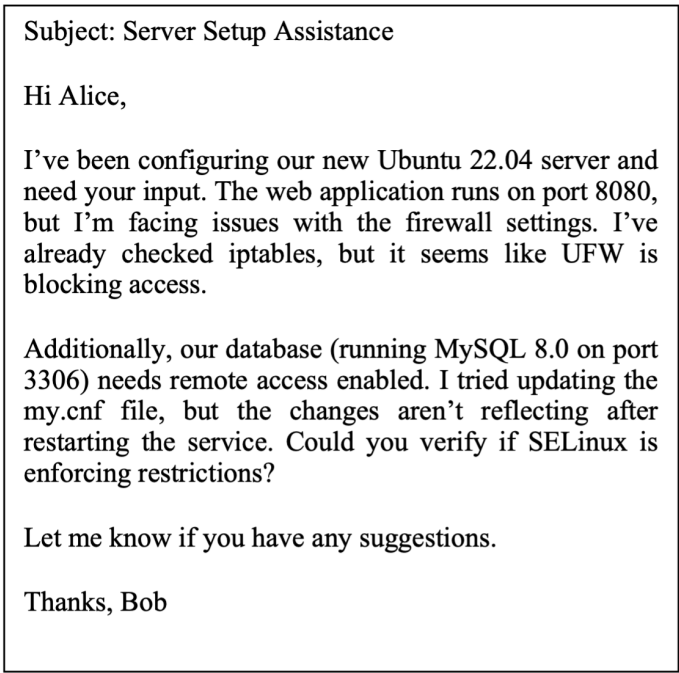


Figure 1. Sample e-mail Format.

Ideally, the system should identify technical terms mentioned in each message, categorize them meaningfully, and determine possible relationships among these elements within the e-mail’s content. Such relationships form the basis for constructing lightweight ontologies or integration layers with configuration management databases (CMDBs). To maintain focus on the problem, the analysis deliberately excludes contextual details such as sender and recipient information, timestamps, attachments, or external references. A sample of the intended structured output is provided in Table 1.

The example illustrates the two central components of the problem: (1) distinguishing technical e-mails from nontechnical ones and (2) extracting structured, actionable insights from technical content. This structured information can be used effectively to build knowl-

edge bases or link to existing ontologies. From a mathematical perspective, the task can be described in two stages. Given a set of e-mails $E = \{e_1, e_2, \dots, e_n\}$, each e-mail e_i is first mapped to a label $y_i \in \{0, 1\}$, where $y_i = 1$ denotes technical relevance. The classification model $f_\theta : E \rightarrow Y$ is trained to minimize the classification loss L_{cls} . For each message labeled as technical, a secondary function extracts entities $\mathcal{T}_i = \{t_1, t_2, \dots, t_m\}$ representing relevant technical components such as software, hardware, or configurations.

Table 1. Expected Structured Output.

Entity	Type	Relation
Ubuntu 22.04	Operating System	Hosts the web application
MySQL 8.0	Database	Runs on port 3306
Port 8080	Network Port	Used by the web application
Firewall	Security Tool	Blocks access to port 8080
SELinux	Security Module	Might be restricting MySQL

One of the main challenges is clearly defining what qualifies as a ‘technical’ e-mail. Although many forms of digital communication contain technical terminology, this study specifically targets messages that reveal concrete details about the IT infrastructure. The Enron dataset includes spam, promotional content, and newsletters that may reference technical terms such as “databases” or “servers” but lack substantive technical discussion. For example, advertisements from software vendors often mention such terms without offering meaningful insight. Likewise, system-generated log messages, while technical, are usually distinguishable by their source and fall outside the scope of our analysis. To operationalize this distinction, technical relevance is defined by the presence of domain-specific entities (software, hardware, protocols, configurations) co-occurring with functional or relational verbs such as “run,” “host,” “connect,” or “install.” In the context of Enron, some e-mails focus on energy systems or industrial hardware domains that are technical but not directly relevant to IT-focused communication and therefore excluded from our primary scope.

Recent advancements in large language models (LLMs) such as OpenAI GPT-4 and DeepSeek-R1 have proven effective for entity extraction, identifying structured information in unstructured or poorly formatted documents. Their generalization across domains and accessible APIs make them useful for rapid prototyping. Performance can be improved through refined prompts or fine-tuning. However, these models are not easily explainable and often require extensive computational resources, which limits their deployment within enterprise environments constrained by data privacy or cost considerations.

While LLMs have demonstrated remarkable versatility, they do not eliminate the need for domain-specific NLP solutions, as task-tailored models often deliver superior performance in specialized contexts. In this study, the identification of technical e-mails is framed as a supervised classification problem, while the recognition of technical entities is treated as a semantic extraction task. Both components are conceptually integrated within a dual-objective framework that links classification accuracy with domain-specific interpretability. Hybrid approaches that combine LLMs with traditional NLP techniques may further balance adaptability and precision, underscoring the importance of continued exploration in this area. This formulation enhances methodological transparency and establishes a rigorous basis for developing reproducible and interpretable NLP models applicable to enterprise communication data.

4. Data Preparation and Preprocessing of the Enron Dataset

We used the 7 May 2015 release of the Enron e-mail dataset provided by Carnegie Mellon University [1], which comprises 517,401 e-mails from 150 Enron employees. To

ensure both reproducibility and methodological consistency, we implemented a structured, multi-step preprocessing pipeline as follows:

4.1. Duplicate Removal

E-mails with identical subject lines and body content were considered duplicates and eliminated. This reduced the dataset size from 517,401 to 245,086 unique e-mails. This step prevents overrepresentation of forwarded or repeated internal announcements, ensuring that classification models are trained on distinct communication instances.

4.2. Length Truncation

To handle outliers, e-mails exceeding 30,000 characters were truncated at this threshold. This limit was empirically chosen after observing that extremely long messages primarily contained logs, disclaimers, or e-mail chains irrelevant to the classification objective.

4.3. Text Normalization

The text normalization process was designed to create a consistent representation of both the e-mail subject and body for downstream modeling.

The following transformations were applied sequentially for the Sense2Vec-based method, which requires tokenization aligned with its pretrained embedding vocabulary:

- The subject and body of each e-mail were converted to lowercase to ensure uniform token matching.
- Line breaks were replaced with underscores (_) to maintain sentence continuity in models expecting single-line inputs.
- Missing subject or body fields were filled with the placeholder string "none" to preserve structural integrity.
- Repetitive character patterns, such as % and =, were removed to minimize encoding artifacts.
- All spaces and newline characters were replaced with underscores (_) for consistent tokenization.
- An underscore character was appended to the beginning and end of each e-mail to facilitate reliable boundary detection during parsing.

Figure 2 illustrates an example of a preprocessed e-mail body after these normalization steps.

```
"body": "_traveling_to_have_a_business_meeting_takes
_the_fun_out_of_the_trip_._especially_if_you_have_to
_prepare_a_presentation_._i_would_suggest_holding_t
he_business_plan_meetings_here_then_take_a_trip_wit
hout_any_formal_business_meetings_. _i_would_even_t
ry_and_get_some_honest_opinions_on_whether_a_trip
_is_even_desired_or_necessary_. _as_far_as_the_busine
ss_meetings_,_i_think_it_would_be_more_productive_t
o_try_and_stimulate_discussions_across_the_different_
groups_about_what_is_working_and_what_is_not_. _to
o_often_the_presenter_speaks_and_the_others_are_qui
et_just_waiting_for_their_turn_. _the_meetings_might_
be_better_if_held_in_a_round_table_discussion_format
_. _my_suggestion_for_where_to_go_is_austin_. _play_
golf_and_rent_a_ski_boat_and_jet_ski_'s_. _flying_so
mewhere_takes_too_much_time_._",
"subject": "_none_"
```

Figure 2. Example of a Sense2Vec-normalized e-mail body.

No underscore normalization or lowercase conversion was applied to the text processed by FLAN-T5-Large. The model received near-raw e-mail text, with standard whitespace preserved to maintain natural-language structure for contextual interpretation.

While punctuation is often removed during normalization, it was deliberately retained in this study due to its potential value for identifying file paths, configuration patterns, and command syntax in technical content.

5. E-Mail Classification with FLAN-T5-Large

The FLAN-T5 model is based on Google's Text-to-Text Transfer Transformer (T5) architecture, which treats all NLP tasks as text-to-text problems. In this paradigm, both input and output are represented as text sequences, allowing a unified framework for tasks such as classification, summarization, and translation. Fine-tuned Language Net (FLAN) extends T5 by applying large-scale instruction tuning, enabling the model to follow natural-language prompts describing task objectives. Through this process, it generalizes to unseen tasks with minimal additional supervision, which is particularly advantageous in enterprise scenarios where labeled data are often limited and heterogeneous.

In this study, e-mails were classified using the FLAN-T5-Large model to balance computational cost and contextual reasoning power. This configuration offers a practical compromise between expressiveness and inference efficiency, supporting scalable deployment in organizational environments. The model is evaluated in its instruction-following configuration to prioritize reproducibility and deployment transparency in enterprise settings.

Although FLAN-T5 was prompted to generate more granular output labels such as "Technical Log," "Technical Spam," and "Technical Newsletter" to enable qualitative inspection and contextual differentiation, these extended categories were produced directly through the instruction prompt, which explicitly allowed the model to generate one of several predefined descriptive labels. The quantitative evaluation, however, was intentionally restricted to a strictly binary formulation. Only the primary "Technical" output was treated as the positive class. All other generated labels—including subcategories and "Unidentified"—were mapped to the Nontechnical class for metric computation. This conservative mapping preserves a clear operational definition of infrastructure-relevant content and ensures consistency in performance evaluation. Consequently, all reported performance metrics correspond to a two-class classification problem (Technical vs. Nontechnical).

A total of 245,086 preprocessed e-mails were classified using FLAN-T5-Large within the Google Colab environment, leveraging an NVIDIA T4 GPU. The implementation was carried out using the Hugging Face Transformers library to ensure transparency and reproducibility. Input sequences were standardized to a fixed length of 512 tokens (approximately 3000 characters) through truncation or padding, maintaining consistent input structure. To mitigate truncation bias, longer messages were processed using a structured policy that preserved complete subject lines, boundary sentences, and domain-relevant expressions such as "server," "port," or "version." This approach ensured that key technical information remained intact within the model's context window.

The model was configured with a decoding temperature of 0 to ensure deterministic outputs. Inference was performed using greedy decoding (`temperature = 0`, `max_length = 512`) without sampling-based strategies (e.g., top-k or top-p). No constrained decoding or vocabulary restriction mechanisms were applied; outputs were generated using the standard `model.generate()` function from the Hugging Face Transformers library. This setup provides stable and auditable predictions, which is crucial for enterprise-grade reproducibility. Each e-mail was processed as a combined subject-body string appended to the following instruction: "Classify the e-mail into one of the following categories: Technical, Technical Log, Technical Spam, Technical Newsletter, Non-Technical. Respond with only one category

name.”. Classification decisions were based directly on the generated categorical label. The model operated under free-form generation. Generated outputs were post-processed and mapped to the predefined category set by resolving minor casing or formatting variations. Outputs that could not be unambiguously mapped were labeled as “Unidentified.” For quantitative evaluation, all outputs were consolidated into a binary taxonomy (Technical vs. Nontechnical) as described above. No explicit probability extraction, token-level scoring, or post hoc calibration procedure was applied. Outputs that did not conform to the predefined label set were retained for qualitative inspection.

As shown in Figure 3, the instruction prompt guided the model to determine each e-mail’s technical relevance. The resulting label distribution across the Enron corpus is summarized in Table 2.

Classify the email into one of the following categories: Technical, Technical Log, Technical Spam, Technical Newsletter, Non-Technical. Respond with only one category name.
Subject: need help with firewall on ubuntu
hi alice, i’ve been configuring our new ubuntu 22.04

Figure 3. Sample instruction template used for FLAN-T5-Large classification.

Table 2. FLAN-T5-Large classification label distribution in the Enron dataset.

Label Category	Number of E-Mails
Technical	28,437
Nontechnical	212,918
Unidentified/Ambiguous	3731

5.1. Evaluation Strategy

To assess model robustness rather than perform parameter training, three *evaluation partitions* were created: (i) a random 80/20 split, (ii) a time-based split where earlier e-mails formed partition A and later ones partition B, and (iii) a cross-mailbox split, in which all messages from specific users were held out for testing. These configurations were designed solely to examine temporal and user-level generalization, as the FLAN-T5-Large model was evaluated under an instruction-following configuration designed to assess generalization across temporal and user-level partitions. This evaluation setup emphasizes robustness and deployment realism rather than parameter optimization.

Table 3 summarizes the evaluation workflow, clarifying how the annotated subset is sampled and partitioned, and indicating which subsets are used for metric computation.

Performance was measured using precision, recall, and macro F1 derived from confusion-matrix counts. All summary metrics were computed on the manually annotated subset described in Section 7. For each evaluation partition, metrics were calculated independently on the corresponding held-out subset. The size of each held-out evaluation subset(n) is explicitly reported in Table 4 to ensure transparency regarding the basis of each estimate.

Although precision remains moderate, recall is stable across all evaluation settings, indicating robustness to temporal and user-level variability. Most false positives originated from vendor newsletters or automated alerts referencing technical terms without substantive discussion. A qualitative inspection further reveals systematic error patterns. False

negatives typically involved brief or implicitly technical exchanges where infrastructure relevance was assumed rather than explicitly articulated, such as short troubleshooting confirmations or configuration acknowledgments. This pattern reflects a common limitation of large instruction-tuned models, which tend to rely on surface lexical cues in ambiguous contexts. In enterprise corpora, contextual grounding is often more nuanced; without domain-specific fine-tuning, the model may assign technical relevance based on recurring keywords or overlook implicitly technical interactions. These findings highlight the potential of hybrid strategies that integrate rule-based post-processing to improve overall precision.

Table 3. Overview of sampling, partitioning, and metric computation workflow.

Stage	Dataset	Size	Purpose	Used for Metrics
Full corpus	Preprocessed Enron e-mails	245,086	Model inference-corpus-level distribution analysis	No
Stratified sampling	Manually annotated subset	1000	Labeled evaluation set	Yes
Random partition	Held-out subset	200	Robustness to sampling variation	Yes
Time-based partition	Held-out subset	500	Temporal robustness	Yes
Cross-mailbox partition	Held-out subset	200	User-level robustness	Yes

Table 4. FLAN-T5-Large classification results across three evaluation partitions.

Evaluation Partition	Size (<i>n</i>)	Precision	Recall	F1
Random 80/20 Partition	200	0.253	0.688	0.368
Time-Based Partition	500	0.241	0.661	0.349
Cross-Mailbox Partition	200	0.229	0.627	0.329

Note: Metrics are computed on held-out subset of the manually annotated data ($n = 1000$); partitions are used solely for evaluation (no model fine-tuning).

Taken together, these findings provide a stable reference point for subsequent comparative evaluation with the Sense2Vec-based approach presented in Section 7.

5.2. Interpretability and Reproducibility

Beyond quantitative performance, interpretability and reproducibility are key to enterprise adoption. Because FLAN-T5 operates on explicit natural-language instructions, all experiments can be replicated through version-controlled prompt templates, enabling traceable and auditable inference pipelines. This ensures transparency and supports the development of domain-adapted NLP systems that can be extended or integrated with other enterprise applications.

In summary, FLAN-T5-Large demonstrates stable recall behavior and methodological consistency across multiple evaluation scenarios. While further domain-specific adaptation could enhance precision, the model provides a reproducible and interpretable reference configuration for enterprise-level NLP classification, addressing both methodological clarity and practical applicability.

6. Semantic Classification and Entity Extraction via Sense2Vec

Sense2Vec is an advanced pretrained word vector model that extends Word2Vec by representing words according to their contextual meanings. Trained on Reddit comments from 2015–2019 [3], it captures semantic relations between words and phrases within a multidimensional vector space. This contextual awareness improves entity representation and benefits key NLP tasks such as NER, keyword extraction, and semantic similarity analysis.

6.1. Keyword Generation and Filtering

To construct a technical keyword space, 79 seed terms were supplied to the Sense2Vec-1g model, generating approximately 100,000 related terms for each input. These terms were stored in a local database for further analysis. Examples of the initial seed terms are shown in Figure 4. The complete seed keyword list is provided in the Appendix A.

```
"web server," "Microsoft," "nginx," "SAP," "Jira," "Zabbix,"
"Nagios," "Syslog," "Wireshark," "Python," "C#," "Power
BI," "Ubuntu," "RDP," "Port," "Metasploit," "Bash,"
"PuTTY," "Azure," "Amazon S3," "Active Directory," "FTP,"
"Server," "Switch," "Firewall," "Palo Alto Networks,"
"cybersecurity," "programming language," "Fortigate," "IP
address," "Visual Studio," "Powershell," "TeamViewer,"
"Outlook," "VMware Workstation," "Hypervisor," "Hyper-V,"
"MySQL," "MSSQL," "PostgreSQL," ".NET," "Shell,"
"Windows Server," "Kerberos," "WSUS," "SharePoint,"
"LDAP," "Antivirus," "Kaspersky," "Symantec," "WMIC,"
"Redis," "DDoS," "Group Policy," "Domain Name," "PHP,"
"JSON," "Skype," "Server Manager," "Android Studio,"
"PyCharm," "Notepad++," "Git," "Eclipse," "Java," "Sublime
Text," "VirtualBox," "Flutter," "NetBeans," "VPN," "Emacs,"
"SIEM," "Java SE," "Vim," "Pandas," "MongoDB"
```

Figure 4. Seed keywords provided to the Sense2Vec model.

Each generated term was assigned an aggregated word frequency (WF_i) that reflected its contextual similarity to the seed term. The cumulative score $F\text{-total}$ was calculated as:

$$F\text{-total} = \sum_{i=1}^N WF_i, \quad (1)$$

where N represents the total number of instances. Higher $F\text{-total}$ values indicate greater semantic relevance. Low-scoring terms were removed to ensure that only high-quality, contextually relevant keywords were retained. Out of approximately 7.9 million initially generated candidates, 658,881 unique terms were preserved after filtering, representing about 8.3% of the total, an expected retention ratio for high-dimensional embedding models trained on noisy, open-domain data.

Filtered terms were clustered using K-Means, and the optimal number of clusters was determined via the Elbow Method. Both token length and semantic similarity were considered. Common or overly generic expressions (frequent one-grams, two-grams, and trigrams) were eliminated based on their frequency in the Enron corpus. Very short tokens (one or two characters) were excluded to improve precision. For instance, ".net" was stored as "_net," and "IP address" as "ip_address." The token "IP" was retained due to its contribution to document-level scoring.

Although some ambiguity persisted, for instance, "apple" could refer to the company or the fruit, these cases had a negligible impact on accuracy. Acronyms such as "ISP" or "IIS" occasionally produced ambiguous matches; however, overall recall improvements outweighed the minor precision losses.

After all filtering steps, 658,881 unique keywords were retained. A selection of the most relevant entries is shown in Table 5.

Relevant terms such as "vscode", "gitlab", and "keepass" emerged even though they were not part of the original seed list, demonstrating the model's ability to infer new domain-specific vocabulary.

Table 5. Generated keywords sorted by *F-total*.

No.	<i>F-total</i>	Keyword
1	569.1320	.net
2	552.7804	vscode
3	528.1036	javascript
4	478.8705	pfsense
5	462.7488	mysql
6	449.9434	firefox
7	434.3514	freenas
8	433.8756	c++
9	433.1822	powershell
10	425.9268	ios
11	423.5753	node.js
12	421.8395	cloudflare
13	413.2999	raspberry_pi
14	401.5831	linux
15	398.6190	rpi
16	398.5853	plex
17	392.1243	keepass
18	387.4339	gitlab
19	383.1945	nginx

6.2. Keyword Matching and Scoring

To apply the extracted keywords to the Enron corpus, the following procedure was used:

- The Aho–Corasick string-matching algorithm [29] detected keyword occurrences efficiently within each e-mail.
- Matched keywords were aggregated and ranked by frequency and contextual relevance.
- Common terms were removed to increase specificity.
- The process was refined iteratively to minimize false positives.

Each e-mail was assigned an aggregated *WF* score computed from all matched keywords, ensuring only one instance per unique term. Scores were normalized across the dataset; e-mails with higher *WF* values were labeled as more technically relevant. Subjects and bodies were analyzed separately. A sample processed e-mail is illustrated in Figure 5.

1. login: pallen pw: ke9davis I don't think these are required by the ISP
2. static IP address IP: 64.216.90.105 Sub: 255.255.255.248 gate: 64.216.90.110 DNS: 151.164.1.8
3. Company: 0413 RC: 105891"

Figure 5. Example of a technical e-mail identified in the Enron dataset.

E-mails with no detected keywords were consistently nontechnical, confirming the scoring system's sensitivity. To refine the classification, auxiliary variables such as e-mail length and keyword count were incorporated to derive a balanced metric for relevance.

6.3. Weighted Scoring and Classification

A weighted score (WS) was introduced to adjust for document length and keyword density:

$$WS = \frac{WF \times (1 + length_n)}{length_n} \quad (2)$$

where $length_n$ denotes the normalized e-mail length. This formulation reduces length-related bias by scaling weighted keyword frequency relative to document size, ensuring that shorter e-mails are not disproportionately penalized while maintaining proportionality for longer texts. No additional length transformation, rescaling, or post hoc normalization was applied beyond the formulation shown in Equation (2).

The decision threshold ($WS = 1.572$) was selected based on inspection of the WS distribution across the full corpus to identify a stable separation between low-density background matches and high-density infrastructure-related content. This threshold corresponds approximately to the upper tail of the WS distribution (about the top 13% of e-mails), resulting in 31,857 messages labeled as technical. Table 6 summarizes the results.

Table 6. Sense2Vec classification results for the Enron dataset.

Label	Count
Nontechnical	213,229
Technical	31,857
Total	245,086

An example of identified entities, including "login" and "ISP", is shown in Figure 6.

```
{
  "message_body": " _1_login_pallen_pw-
_ke9davis_i_don_t_think_these_are_required_by_the_isp_2
_static_ip_address_ip_64_216_90_105_sub-
_255_255_255_248_gate_6_4_216_90_110_dns-
_151_164_1_83_company_0413_rc_105891.",
  "body_match": [
    {
      "name": " _login_",
      "start": 4,
      "end": 11,
      "len": 7,
      "duplicate": 0,
      "f": "0.28523681159824305"
    },
    {
      "name": " _isp_",
      "start": 75,
      "end": 80,
      "len": 5,
      "duplicate": 0,
      "f": "0.2270580375334327"
    }
  ],
  "f_total": 1.4974011422880658
}
```

Figure 6. Example e-mail containing matched entities detected by Sense2Vec.

6.4. Clustering and Visualization

Score-based clustering was used to group e-mails based on total relevance, length, and hit density. The resulting heatmap in Figure 7 highlights the separation achieved by this method.

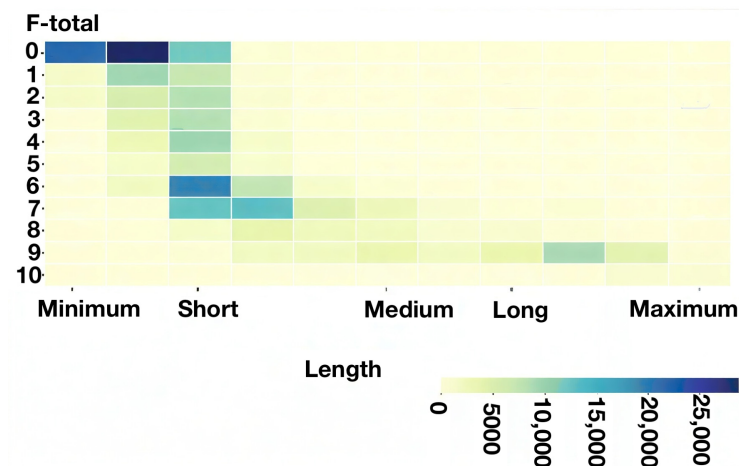


Figure 7. Heatmap of keyword density by e-mail length and *F-total*.

As shown in Figure 7, short e-mails generally exhibit lower *F-total* values, indicating limited technical content, whereas longer e-mails show higher variability and greater relevance. A concentrated cluster at *F-total* = 0 represents nontechnical correspondence, effectively filtering noise and reducing the dataset for downstream analysis.

Overall, the combined use of *F-total* and *WS* metrics effectively separates technical from nontechnical e-mails, minimizing misclassifications and significantly reducing manual review effort. Entity recognition accurately identifies relevant components, with remaining ambiguities (e.g., acronyms or multi-domain terms) presenting opportunities for future refinement through contextual fine-tuning.

Although some false positives emerged—particularly from contextually vague tokens—the trade-off was deemed acceptable to ensure high coverage. In exploratory settings, coverage-oriented strategies enable discovery of emerging terminology and weakly expressed infrastructure references, which are often missed by strict classifiers. In future extensions, post-processing or disambiguation modules can further refine precision without reducing semantic coverage.

7. Results

The following results are presented to illustrate complementary behaviors and deployment trade-offs between contextual and semantic approaches, rather than to establish model superiority. This section presents the experimental results obtained from both the FLAN-T5-Large and Sense2Vec-based models across the Enron e-mail dataset. The analyses include comparative evaluations between the two classifiers (T5Class and SClass), as well as quantitative performance validation using a manually annotated subset of 1000 e-mails. The reported metrics—accuracy, precision, recall, and F1—reflect each model’s ability to distinguish technical from nontechnical content under varying evaluation settings. The section is organized as follows: first, a direct comparison of model outputs is provided; next, model performances are analyzed on the annotated dataset to assess their consistency, generalization, and classification reliability. Overall, the results establish a clear trade-off between precision and recall, with FLAN-T5-Large achieving higher accuracy and Sense2Vec offering greater semantic coverage.

7.1. Cross-Analysis of FLAN-T5 and Sense2Vec Outputs

To evaluate the consistency between the two models, we compared Sense2Vec-based model predictions with the classification results of the FLAN-T5-based model. Among a total of 245,086 e-mails, both models concurred in identifying 203,615 e-mails as nontechnical. However, the Sense2Vec-based model classified an additional 28,605 e-mails as technical,

which FLAN-T5 had previously categorized as nontechnical. A detailed comparison is presented in Table 7. To maintain clarity, the classifier based on FLAN-T5 is referred to as T5Class, and the one based on Sense2Vec as SClass throughout this section.

Table 7. Cross-analysis of FLAN-T5-based and Sense2Vec-based classifications.

T5Class vs. SClass	Nontechnical (S)	Technical (S)
Nontechnical	203,615	28,605
Technical	7510	2669
Technical Log	162	39
Technical Newsletter	408	76
Technical Spam	1389	377
Unidentified	145	91

To validate the classification consistency, the highest-ranked e-mails were inspected alongside their scores. Table 8 lists the subjects of the top 25 e-mails that both models classified as technical, sorted by their relevance scores.

Table 8. Top 25 technical e-mails classified by both models.

No.	E-Mail Subject	Score
1	terminal_server_	1.8231
2	workstation_software_updates_	1.8201
3	db_outage	1.8201
4	unsure_of_who_to_call_	1.8190
5	quiz_solutions_	1.8190
6	west_pipeline_daily_morning_report_	1.8153
7	installation_for_full_requirement_dispatch_application_	1.8151
8	good_morning_	1.8087
9	answer_for_logging_in_from_home_	1.8087
10	pdesk_in_stage_	1.8087
11	fyi_:taqq/_erms_environment_	1.8081
12	checkout_for_5/_3_hourly_deals_	1.8067
13	windows_login_	1.8057
14	west_pl_daily_morning_report_	1.8047
15	server_error_	1.8043
16	fist_logins_	1.8022
17	new_version_of_netscape_	1.8014
18	livelink_access_	1.8014
19	global_standards_database_mtc_instructions	1.8014
20	ipayit_system_outage_	1.8012
21	windows_login_.	1.8004
22	popsprd_migration_	1.7992
23	start_date_: 4/_8/_01_: hourahead_hour_:_10_: <codesite>_	1.7988
24	finding_the_unify_gas_menu_item_	1.7985
25	project_tracking_database_	1.7981

Both models exhibited alignment on clearly technical subjects (e.g., “server error,” “system outage,” “database instructions”) but diverged in cases involving mixed or context-dependent phrases. This suggests that while FLAN-T5’s contextual reasoning provides more stable judgments, Sense2Vec captures a broader but noisier term space.

7.2. Performance on the Annotated Dataset

To quantitatively assess model performance, both T5Class and SClass were evaluated on a manually annotated subset of 1000 e-mails, consisting of 905 nontechnical and

95 technical instances (Table 9). The annotation followed the same labeling scheme used in prior evaluations.

Table 9. Annotated label distribution in the evaluation subset.

Label	Sample Count
Nontechnical	905
Technical	95
Total	1000

The annotation process followed a deterministic protocol based on domain-specific patterns, including the co-occurrence of infrastructure-related entities (e.g., server, database, port) with operational verbs (e.g., install, run, host). Messages were labeled based on their functional relevance to enterprise IT systems, rather than merely the presence of technical terms. This ensured that the evaluation set represented both high-confidence technical content and edge cases that required contextual interpretation. The annotated subset was labeled by a single domain-informed annotator following this deterministic protocol. Because the labeling criteria were explicitly rule-based and consistently applied, inter-annotator agreement statistics are not reported in this study.

The FLAN-T5-based classifier achieved the results summarized in Table 10. From these, TP = 24, FP = 71, FN = 36, and TN = 869 were derived, yielding the metrics shown in Table 11.

Table 10. Annotated label classification results of the FLAN-T5-based model (T5Class).

Predicted Label	Actual Label	Result	Sample Count
Nontechnical	Nontechnical	Success	866
Nontechnical	Technical	Fail	69
Technical	Nontechnical	Fail	36
Technical	Technical	Success	24
Technical Newsletter	Nontechnical	Success	1
Technical Spam	Nontechnical	Success	13
Technical Spam	Technical	Fail	2
Unidentified	Nontechnical	Fail	1

Table 11. Performance metrics for the FLAN-T5-based model (T5Class).

Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
90.4	25.3	40.0	31.0

The high accuracy is primarily due to the dominance of nontechnical samples. Nevertheless, precision (25.3%) and recall (40.0%) reveal that the model, while precise, tends to under-detect borderline technical content.

The Sense2Vec-based classifier (SClass) exhibited comparatively higher recall but lower precision, as summarized in Tables 12 and 13.

Table 12. Annotated label classification results of the Sense2Vec-based model (SClass).

Predicted Label	Actual Label	Result	Sample Count
Nontechnical	Nontechnical	Success	786
Nontechnical	Technical	Fail	66
Technical	Nontechnical	Fail	119
Technical	Technical	Success	29

Table 13. Performance metrics for the Sense2Vec-based model (SClass).

Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
81.5	19.6	30.5	23.8

The FLAN-T5-Large model achieved higher overall accuracy (90.4%), outperforming the Sense2Vec-based approach in most metrics. However, the latter displayed a more balanced recall–precision trade-off, which could be advantageous in exploratory applications that favor inclusiveness over strict correctness. Adjusting decision thresholds or incorporating weighted confidence scoring could further improve both precision and recall in future iterations.

A closer inspection of misclassified instances helps explain this trade-off. In the Sense2Vec-based classifier, false positives were primarily associated with lexical over-expansion, where semantically related but contextually irrelevant terms triggered technical classification. Conversely, false negatives often occurred when technical relevance depended on contextual cues rather than explicit keyword presence. These patterns reflect the inherent differences between distributional semantic matching and instruction-based contextual reasoning. Together, these findings clarify the structural trade-offs between contextual inference and semantic expansion strategies in enterprise e-mail classification.

8. Discussion

8.1. Interpretation of Results and Practical Implications

From an application perspective, the proposed approach is relevant to several enterprise use cases, including IT incident triage, internal audit support, and the structuring of organizational knowledge bases derived from unstructured e-mail communication.

The results suggest that contextual LLM-based classification and lightweight semantic methods can play complementary roles within enterprise NLP pipelines, depending on operational priorities. The comparative evaluation confirms that both the FLAN-T5-Large and Sense2Vec-based models effectively filter nontechnical content from enterprise e-mail corpora. The FLAN-T5 model, leveraging instruction-tuned contextual reasoning, consistently demonstrates higher accuracy and precision. This indicates its suitability for high-confidence use cases such as IT operations, audit, or cybersecurity workflows where false positives must be minimized. In contrast, the Sense2Vec model identifies a broader spectrum of potentially relevant e-mails, prioritizing recall over precision, and therefore serves better in exploratory or research-oriented contexts.

These results emphasize the necessity of model selection based on operational priorities. Precision-focused environments may benefit from FLAN-T5’s deterministic outputs, while Sense2Vec provides value in enriching keyword databases or identifying emerging terminology. A hybrid configuration—where FLAN-T5 filters candidate e-mails and Sense2Vec expands their semantic footprint—could potentially yield improved balance and overall robustness. Building on this complementarity, a practical configuration involves deploying FLAN-T5 as a high-precision filter to identify core technical content, followed by Sense2Vec-based enrichment to expand lexical coverage and capture peripheral or contextually weak indicators. This layered design ensures both audit-ready accuracy and exploratory breadth, making it suitable for enterprise workflows such as incident triage, knowledge base population, or post-mortem analysis pipelines.

Moreover, the analysis of the annotated subset underscores the importance of accounting for class imbalance. Accuracy alone can be misleading in datasets where nontechnical content predominates. In such cases, recall and F1 provide a more accurate reflection of

classification quality. Hence, future comparative studies should consistently report all major metrics to ensure fair assessment.

8.2. Limitations and Future Work

While the proposed framework achieves reliable and interpretable results across large-scale e-mail corpora, several aspects offer opportunities for further refinement rather than fundamental limitations. Ambiguity in acronyms and context-dependent entities occasionally introduces minor noise in classification, which could be mitigated by integrating disambiguation modules based on contextual embeddings or external knowledge graphs. Likewise, improving recall for shorter or partially technical e-mails represents a natural next step, potentially through dynamic threshold optimization or domain-specific adaptation strategies applied to the underlying language models. While the precision of both models appears modest, this is consistent with the challenging nature of enterprise e-mail corpora, where partial technical language and ambiguous context are common. The models were deliberately evaluated in an instruction-following configuration to prioritize reproducibility, interpretability, and application flexibility across organizational settings. Future configurations may adopt dynamic thresholds or ensemble fusion to better balance precision and recall based on organizational priorities.

In addition, this study does not include supervised baseline models (e.g., TF-IDF combined with logistic regression or fine-tuned transformer architectures trained on the annotated subset). The primary objective was to compare two deployment-ready approaches that operate without task-specific fine-tuning, reflecting enterprise environments where labeled data and model training resources may be limited. Incorporating supervised baselines would shift the methodological focus from non-fine-tuned operational comparison to supervised learning benchmarking. Future research may extend the present framework by introducing supervised baselines under controlled cross-validation settings.

Although large-scale inference was practically feasible using a single NVIDIA T4 GPU in batch mode, detailed latency and throughput benchmarking under varying hardware and deployment configurations was beyond the scope of this study. Future work may systematically evaluate computational efficiency and cost-performance trade-offs in production-scale enterprise environments.

Another promising direction involves adaptability and scalability. As enterprise terminology evolves continuously, incorporating incremental learning or active feedback mechanisms can ensure long-term robustness without retraining from scratch. Extending the framework to capture discourse-level relationships, such as intent shifts across e-mail threads, may also enhance semantic granularity.

Finally, the efficiency of large-scale deployment can be further improved through lightweight adaptation and post-processing strategies, ensuring scalability under realistic organizational constraints. Overall, the presented approach establishes a strong and scalable foundation for automated enterprise communication analysis, offering multiple pathways for continued research and industrial adoption.

In summary, this study delivers a practical and reproducible framework for detecting technical e-mails in enterprise environments through two complementary paradigms, contextual instruction, following (FLAN-T5-Large) and semantic embedding (Sense2Vec). Together, they highlight the balance between precision-driven contextual models and coverage-oriented lexical representations. The findings demonstrate not only the feasibility of automated technical content detection at scale but also its relevance to broader NLP applications such as IT incident management, support automation, and knowledge extraction.

The proposed methodology therefore represents a significant step toward scalable and interpretable enterprise NLP solutions. With targeted refinements in model adaptability

and domain calibration, this framework can evolve into an operational tool for intelligent document classification and contextual understanding in real-world organizational settings. The comparative analysis demonstrated that domain-informed heuristics and prompt-based instruction tuning can achieve reproducible, explainable, and semantically rich classification over legacy enterprise corpora. By aligning the evaluation strategy with operational enterprise constraints, this study provides a deployable and extensible foundation for real-world NLP applications.

Author Contributions: Conceptualization, A.Ç.S. and Ş.D.-O.; methodology, A.Ç.S. and Ş.D.-O.; software, A.Ç.S.; validation, A.Ç.S.; formal analysis, A.Ç.S.; investigation, A.Ç.S.; data curation, A.Ç.S.; writing—original draft preparation, A.Ç.S. and Ş.D.-O.; writing—review and editing, A.Ç.S. and Ş.D.-O.; visualization, A.Ç.S.; supervision, Ş.D.-O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Enron e-mail dataset is publicly available from Carnegie Mellon University. Derived materials such as keyword lists and annotation scripts can be provided by the corresponding author upon request. Please send your inquiries to safak.odabasi@iuc.edu.tr.

Acknowledgments: This article is derived from the M.Sc. thesis titled ‘*Natural Language Processing Applications for Cybersecurity*’, completed at Bahçeşehir University, Department of Cybersecurity.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

API	Application Programming Interface
AMP	Advanced Messaging Platform
CMDB	Configuration Management Database
CNN	Convolutional Neural Network
FLAN	Fine-Tuned Language Net
FLAN-T5	Fine-Tuned Language Net based on Text-to-Text Transfer Transformer
FN	False Negative
FP	False Positive
F1	Harmonic Mean of Precision and Recall
GPU	Graphics Processing Unit
IT	Information Technology
K-Means	K-Means Clustering Algorithm
LLM	Large Language Model
LSTM	Long Short-Term Memory
ML	Machine Learning
NER	Named Entity Recognition
NLP	Natural Language Processing
RNN	Recurrent Neural Network
SClass	Sense2Vec-Based Classifier
T5Class	FLAN-T5-Based Classifier
TN	True Negative
TP	True Positive
WF	Word Frequency
WS	Weighted Score

Appendix A

This appendix provides the complete seed keyword vocabulary used for the Sense2Vec-based semantic expansion approach. The seed list was constructed to cover major enterprise IT infrastructure layers, including operating systems, networking, virtualization, security, monitoring, databases, development environments, enterprise software ecosystems, and hardware components. These seed terms served as anchors for distributional expansion using pretrained Sense2Vec embeddings.

Table A1. Seed keyword vocabulary used for Sense2Vec-based semantic expansion.

Keyword Category	Seed Terms
Operating Systems	Windows, Linux, Ubuntu, macOS, Windows Server 2003
Cloud & Virtualization	Azure, AWS, VMware, Hypervisor, Hyper-V, VirtualBox
Networking & Infrastructure	Port, IP address, Switch, Firewall, Palo Alto, Fortigate, Domain Name, FTP, RDP, Syslog
Monitoring & Observability	Paessler PRTG, Zabbix, Nagios, Wireshark
Directory & Identity Services	Active Directory, LDAP, Kerberos, Group Policy
Server & Web Infrastructure	Server, Internet Information Services, WSUS, SharePoint
Security	Antivirus, Kaspersky, Symantec, DDoS
Databases & Data Systems	MySQL, MSSQL, MongoDB, Redis, SQL Server Manager
Programming Languages & Frameworks	Python, C#, Java, PHP, Flutter, .NET, JSON, Pandas
Development Environments & Tools	Visual Studio, Android Studio, PyCharm, Eclipse, NetBeans, Spyder, RStudio, Sublime Text, Notepad++, Emacs, Vim, Jupyter, Anaconda, Git
Remote Access & Administration Tools	AnyDesk, PuTTY, TeamViewer, PowerShell, Bash, Shell
Enterprise & Business Software	SAP, Microsoft, Excel, Teams, Outlook, Skype, Chrome, Firefox
Hardware & Enterprise Equipment	HP ProLiant

References

1. Enron e-Mail Dataset. Carnegie Mellon University. Available online: <https://www.cs.cmu.edu/~enron/> (accessed on 25 September 2025).
2. Klimt, B.; Yang, Y. The Enron Corpus: A New Dataset for e-mail Classification Research. In *Proceedings of the 15th European Conference on Machine Learning (ECML 2004), Pisa, Italy, 20–24 September 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 217–226.
3. Trask, A.; Michalak, P.; Liu, J. Sense2Vec: A fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv* **2015**, arXiv:1511.06388.
4. Chung, H.W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, E.; Wang, X.; Wang, Z.; Zhou, Y.; et al. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.* **2024**, *25*, 1–53.
5. Janez-Martino, F.; Alaiz-Rodriguez, R.; Gonzalez-Castro, V.; Fidalgo, E.; Alegre, E. Spam e-mail classification based on cybersecurity potential risk using natural language processing. *Knowl.-Based Syst.* **2025**, *310*, 112939. [\[CrossRef\]](#)
6. Filali, A.; Alaoui, E.A.A.; Merras, M. Enhancing spam detection with GANs and BERT embeddings: A novel approach to imbalanced datasets. *Procedia Comput. Sci.* **2024**, *236*, 420–427. [\[CrossRef\]](#)
7. Nasreen, G.; Khan, M.M.; Younus, M.; Zafar, B.; Hanif, M.K. Email spam detection by deep learning models using novel feature selection technique and BERT. *Egypt. Inform. J.* **2024**, *26*, 100473. [\[CrossRef\]](#)
8. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. *ACM Comput. Surv.* **2009**, *41*, 1–58. [\[CrossRef\]](#)
9. Buczak, A.L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1153–1176. [\[CrossRef\]](#)
10. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 649–657.

11. Alshingiti, Z.; Alaqel, R.; Al-Muhtadi, J.; Haq, Q.E.U.; Saleem, K.; Faheem, M.H. A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. *Electronics* **2023**, *12*, 232. [\[CrossRef\]](#)
12. Atawneh, S.; Aljehani, H. Phishing e-mail detection model using deep learning. *Electronics* **2023**, *12*, 4261. [\[CrossRef\]](#)
13. Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 50–70. [\[CrossRef\]](#)
14. Sammet, J.; Krestel, R. Domain-specific keyword extraction using BERT. In Proceedings of the 4th Conference on Language, Data and Knowledge, Vienna, Austria, 12–15 September 2023; pp. 659–665.
15. Shang, J.; Liu, J.; Jiang, M.; Ren, X.; Voss, C.R.; Han, J. Automated phrase mining from massive text corpora. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1825–1837. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Zhang, X.; Li, Q.; Li, C.; Chen, H. Conphrase: Enhancing context-aware phrase mining from text corpora. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 6767–6783. [\[CrossRef\]](#)
17. Altuncu, E.; Nurse, J.R.; Xu, Y.; Guo, J.; Li, S. Improving performance of automatic keyword extraction (AKE) methods using PoS tagging and enhanced semantic-awareness. *Information* **2025**, *16*, 601. [\[CrossRef\]](#)
18. Khan, M.Q.; Shahid, A.; Uddin, M.I.; Roman, M.; Alharbi, A.; Alosaimi, W.; Alshahrani, S.M. Impact analysis of keyword extraction using contextual word embedding. *PeerJ Comput. Sci.* **2022**, *8*, e967. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Yu, B.; Hu, Y.; Mang, Q.; Hu, W.; He, P. Automated testing and improvement of named entity recognition systems. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, San Francisco, CA, USA, 3–9 December 2023; pp. 883–894.
20. Labonne, M.; Moran, S. Spam-T5: Benchmarking large language models for few-shot e-mail spam detection. *arXiv* **2023**, arXiv:2304.01238.
21. Patel, H.; Rehman, U.; Iqbal, F. Evaluating the efficacy of large language models in identifying phishing attempts. In Proceedings of the 16th International Conference on Human System Interaction (HSI 2024), Paris, France, 8–11 July 2024; pp. 1–7.
22. Altwaijry, N.; Al-Turaiki, I.; Alotaibi, R.; Alakeel, F. Advancing phishing e-mail detection: A comparative study of deep learning models. *Sensors* **2024**, *24*, 2077. [\[CrossRef\]](#)
23. Tarapiah, S.; Abbas, L.; Mardawi, O.; Atalla, S.; Himeur, Y.; Mansoor, W. Evaluating the effectiveness of large language models (LLMs) versus machine learning (ML) in identifying and detecting phishing e-mail attempts. *Algorithms* **2025**, *18*, 599. [\[CrossRef\]](#)
24. Reddy, G.P.; Dsouza, R.; Ramyashree, S.R.; Raghavendra, S.; Venugopala, P.S. An LLM driven framework for email spam detection using DistilBERT embeddings and neural classifiers. *Discov. Appl. Sci.* **2026**, *8*, 146. [\[CrossRef\]](#)
25. Kaur, S.; Smiley, C.; Ramani, K.; Kochkina, E.; Sibue, M.; Mensah, S.; Totis, P.; Tilli, C.; Aguda, T.; Borrajo, D.; et al. Advanced Messaging Platform (AMP): Pipeline for Automated Enterprise Email Processing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, Vienna, Austria, 27 July–1 August 2025; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 813–825. [\[CrossRef\]](#)
26. Kumar, R.; Ishan, K.; Kumar, H.; Singla, A. LLM-Powered Knowledge Graphs for Enterprise Intelligence and Analytics. *arXiv* **2025**, arXiv:2503.07993. [\[CrossRef\]](#)
27. Mariotti, L.; Guidetti, V.; Mandreoli, F.; Belli, A.; Lombardi, P. Combining large language models with enterprise knowledge graphs: A perspective on enhanced natural language understanding. *Front. Artif. Intell.* **2024**, *7*, 1460065. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Brachman, M.; El-Ashry, A.; Dugan, C.; Geyer, W. Current and Future Use of Large Language Models for Knowledge Work. *Proc. ACM Hum.-Comput. Interact.* **2025**, *9*, CSCW222. [\[CrossRef\]](#)
29. Aho, A.V.; Corasick, M.J. Efficient string matching: An aid to bibliographic search. *Commun. ACM* **1975**, *18*, 333–340. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.