Label-semantics Aware Generative Approach for Domain-Agnostic Multilabel Classification

Subhendu Khatuya, Shashwat Naidu, Saptarshi Ghosh, Pawan Goyal, Niloy Ganguly
Indian Institute of Technology Kharagpur, India
subha.cse143@gmail.com, shashwatnaidu07@gmail.com,
saptarshi@cse.iitkgp.ac.in, pawang@cse.iitkgp.ac.in, niloy@cse.iitkgp.ac.in

Abstract

The explosion of textual data has made manual document classification increasingly challenging. To address this, we introduce a robust, efficient domain-agnostic generative model framework for multi-label text classification. Instead of treating labels as mere atomic symbols, our approach utilizes predefined label descriptions and is trained to generate these descriptions based on the input text. During inference, the generated descriptions are matched to the predefined labels using a finetuned sentence transformer. We integrate this with a dual-objective loss function, combining cross-entropy loss and cosine similarity of the generated sentences with the predefined target descriptions, ensuring both semantic alignment and accuracy. Our proposed model LAGAMC stands out for its parameter efficiency and versatility across diverse datasets, making it well-suited for practical applications. We demonstrate the effectiveness of our proposed model by achieving new state-of-the-art performances across all evaluated datasets, surpassing several strong baselines. We achieve improvements of **13.94%** in Micro-F1 and **24.85%** in Macro-F1 compared to the closest baseline across all datasets.

1 Introduction

Text classification automates the analysis and organization of large datasets, enabling efficient labeling, categorization, and valuable insights. In text classification, two main categories arise: multi-class and multi-label classification. Multi-class classification assigns a single category to a text, while multi-label classification (MLTC) (Xiao et al., 2019) assigns multiple relevant labels to a document. Real-world applications include topic recognition (Yang et al., 2016), question answering (Kumar et al., 2016), sentiment analysis (Cambria et al., 2014), information retrieval (Gopal and Yang, 2010), and text categorization (Schapire and Singer, 2000), among others.

In the field of multi-label text classification, various approaches including traditional machine learning, deep learning, and hybrid models have been proposed (Chen et al., 2022). Several state-of-the-art methods (Huang et al., 2021; Xiao et al., 2019) emphasize the importance of label descriptions or metadata in improving model performance and capturing label correlations. However, these approaches often face limitations in their generalizability and adaptability. Some models rely on label hierarchies or meta-path graphs (Ye et al., 2024), which, although effective in certain contexts, hinder scalability and flexibility. Additionally, some models are designed for specific tasks, such as legal or financial applications, integrating label descriptions for specialized classifiers (Chalkidis et al., 2020; Khatuya et al., 2024). Despite their advancements, these methods remain limited to particular domains, highlighting the need for a more generalizable approach to MLTC.

Large Language Models (LLMs), with their extensive pretraining, are capable of understanding similarities and relationships based on textual patterns (Naveed et al., 2024). This motivated us to propose a novel domain-agnostic pipeline that leverages recent generative models in a parameter-efficient setup employing Low-Rank Adaptation (LoRA) (Hu et al., 2021), for the MLTC task. Unlike previous approaches that treat label descriptions as metadata, our proposed approach LAGAMC trains the generative model to generate these descriptions directly. This enables us to harness the full potential of LLMs, providing a more nuanced representation of label relationships and their connections to the corresponding text.

Obtaining label descriptions manually for multi-label datasets is both labor-intensive and subjective. To overcome this, we automate the process by leveraging GPT-3.5 (Brown et al., 2020) and Wikipedia, enabling the efficient creation of label descriptions with minimal human intervention. By leveraging

Findings of the Association for Computational Linguistics: ACL 2025, pages 22286–22298

a generative model while providing semantic information about labels, our method consistently outperforms state-of-the-art models, achieving a **9.32%** improvement in Micro-F1 and **15.25%** in Macro-F1 over the closest baseline.

We further integrate a dual-objective loss function, combining cross-entropy loss with a cosine similarity-based loss. This results in an additional increase of **4.62%** and **9.60%** in Micro-F1 and Macro-F1 respectively. Using such a hybrid loss helps bring the embeddings of the outputs of the generative model and that of the predefined label description closer in the representation space, making it easier to map the generated outputs with final labels. This hybrid approach ensures the model comprehends label distinctions and avoids overfitting to token-level matches, contributing to improved performance across diverse datasets.

Empirical evaluations on diverse datasets spanning social media, news, academic, and healthcare domains demonstrate the effectiveness of our approach LAGAMC. Across various datasets, our method consistently outperforms state-of-the-art models, achieving an overall improvement of **13.94%** in Micro-F1 and **24.85%** in Macro-F1 over the closest baseline. Additionally, our model shows strong performance on rare labels and exhibits zero-shot capability, further enhancing its applicability. In summary, the key contributions of this paper are as follows:

- We generated label descriptions for datasets lacking them, using an annotation process guided by GPT-3.5 (Brown et al., 2020) and Wikipedia. Our dataset[1] and code are publicly available at this anonymous link[2].
- We developed a novel parameter-efficient generative approach for MLTC, leveraging label descriptions to improve classification accuracy.
- We introduced a dual-objective loss function, incorporating a semantic similarity-based loss to enhance the model's semantic understanding.
- Our method generalizes well across domains, as demonstrated by performance metrics on various datasets. For all the datasets, our proposed model LAGAMC, achieves huge improvements over baselines. Furthermore, our model excels on rare labels and demonstrates zero-shot capability.

[1]Data: https://drive.google.com/drive/folders/1nrCKgmEtYrM1mQHIu3-eKXEtRdlIUDiO?usp=sharing

[2]Code: https://anonymous.4open.science/r/GenerativeMultiLabel_Classification-5415/README.md

2 Related Works

Multi-label text classification has been approached through a range of techniques, including extending single-label classifiers, employing neural network architectures, and, more recently, utilizing transformer and pretrained language models based works. Neural network-based approaches have shown great success, with methods leveraging CNNs (Liu et al., 2017), RNNs (Liu et al., 2016), and hybrid CNN-RNN models (Chen et al., 2017). Attention-based models (Yang et al., 2016; You et al., 2019; Adhikari et al., 2019) have also improved document representation, but often overlook label semantics and dependencies.

More recent approaches utilize pretrained transformers such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), which have been adapted for multi-label tasks. For instance, (Ameer et al., 2023) applies multiple attention layers over RoBERTa's final layer to enhance label correlation learning. (Ma et al., 2023) explores various loss functions designed to mitigate the impact of class imbalance in datasets.

Most of these do not leverage the information contained in label descriptions. Some of the works that utilize label descriptions to enhance performance (Ye et al., 2024) are not scalable as they require extra information like label hierarchy. Additionally, some models are built for specific tasks (Khatuya et al., 2024), making it difficult to generalize to datasets from different domains.

There also exists popular extreme multi-label classification frameworks like GalaXC (Saini et al., 2021), SiameseXML (Dahiya et al., 2021a), DEXA (Dahiya et al., 2023), Renee (Jain et al., 2023), InceptionXML (Kharbanda et al., 2024) etc. These works focus more on efficiency given the large set of labels to predict from, but rarely utilize label descriptions to enhance performance.

3 Datasets and Label Description Generation

We curated popular multi-label classification datasets from social media, news, academic, and healthcare domains for our experiments.

3.1 Overview of Datasets

CAVES [Social Media]: The CAVES dataset (Poddar et al., 2022) contains 10K anti-vaccine tweets related to COVID-19, labeled manually.
Reuters [Newswire]: Reuters-21578 (Hayes and Weinstein, 1990) consists of documents from the 1987 Reuters newswire, with a skewed distribution.
AAPD [Academic Text]: The Arxiv Academic Paper Dataset (AAPD) (Yang et al., 2018) contains abstracts from the computer science domain.
SemEval [Social Media]: SemEval-2018 Task 1C (Mohammad et al., 2018) includes emotion-labeled tweets from 2016-2017 in English.
PubMed [Healthcare]: A processed version of the PubMed dataset[3] from BioASQ 9 Task A[4], available on Kaggle[5], manually annotated with Medical Subject Headings (MeSH) by biomedical experts.

3.2 Dataset Enhancement with Label Descriptions

Many existing datasets in multi-label classification tasks provide only the final label predictions without offering detailed descriptions for each label. However, label descriptions are essential for improving contextual understanding and enhancing model performance. In this study, we address this gap by augmenting the datasets with refined label descriptions. The CAVES dataset (Poddar et al., 2022), from the social media domain, is the only dataset in our study that already includes this.

For the remaining datasets, we generated label descriptions using GPT-3.5 (Brown et al., 2020) in combination with predefined Wikipedia definitions. Initially, we retrieved label definitions from Wikipedia, referred to as "initial descriptions." To better align these definitions with the specific context of each dataset, we refined them by providing GPT-3.5 with the initial Wikipedia definition and two relevant examples from the dataset where the label appeared in the predictions. The model was then prompted to generate a more contextually appropriate description for each label, ensuring better alignment with the dataset's context.

For example, the following prompt was used for the label '**Anger**' in the SemEval dataset:
Label: *Anger*
Initial Description: *Anger, emotion that involves annoyance and rage.*
Dataset: *Contains tweets and corresponding emotion annotations.*
Examples from the dataset:
Tweet 1: "Tears and eyes can dry but I won't, I'm burning like the wire in a lightbulb."
Prediction: Anger
Tweet 2: "We're going to get City in the next round for a revenge."
Prediction: Anger
Task: *Generate a suitable label description for 'Anger' that fits the context of this dataset.*

Dataset	Train	Dev	Test	# Labels	Max Labels	Avg. Desc. Length
CAVES	6,957	987	1,977	12	3	28.17
SemEval	6,838	886	3,259	12	6	61.11
Reuters	6,769	1,000	3,019	90	11	13.41
AAPD	53,840	1,000	1,000	54	8	50.34
PubMed	40,000	10,000	10,000	14	13	91.40

Table 1: Dataset statistics. The last three columns show the total number of labels, the maximum number of labels per sample, and the average label description length for each dataset.

Evaluation Metrics: To evaluate the performance of our model, we consider 1) Micro-F1 and 2) Macro-F1 metrics.

4 Baselines

We validate our model with different baselines ranging from traditional RNN and CNN based approaches like **TextCNN** (Kim, 2014), **TextRNN** (Liu et al., 2016), **Attentive ConvNet** (Yin and Schütze, 2018) to transformer based approaches like **BERT** (Devlin et al., 2019), **XLNet** (Yang et al., 2020), **RoBERTa** (Liu et al., 2019), **Star-Transformer** (Guo et al., 2022). We also compared the performance of LAGAMC against various popular extreme multi-label classification frameworks like **(AttentionXML** (You et al., 2019), **(GalaXC** (Saini et al., 2021) , **SiameseXML** (Dahiya et al., 2021a), **DEXA** (Dahiya et al., 2023), **DeepXML** (Dahiya et al., 2021b) and **Renee** (Jain et al., 2023).

We also created our own generative baselines **T5-Base** (Raffel et al., 2020) and **T5-Large** (Raffel et al., 2020). Lastly, with the emergence of **ChatGPT** (Brown et al., 2020), we were curious to check it's performance for this task using the same instruction prompt for 500 random samples using *gpt-3.5-turbo* [6] API.

[3] https://pubmed.ncbi.nlm.nih.gov/
[4] http://participantsarea.bioasq.org/general_information/Task9a/
[5] https://t.ly/FWWuQ
[6] https://platform.openai.com/docs/models/gpt-3-5

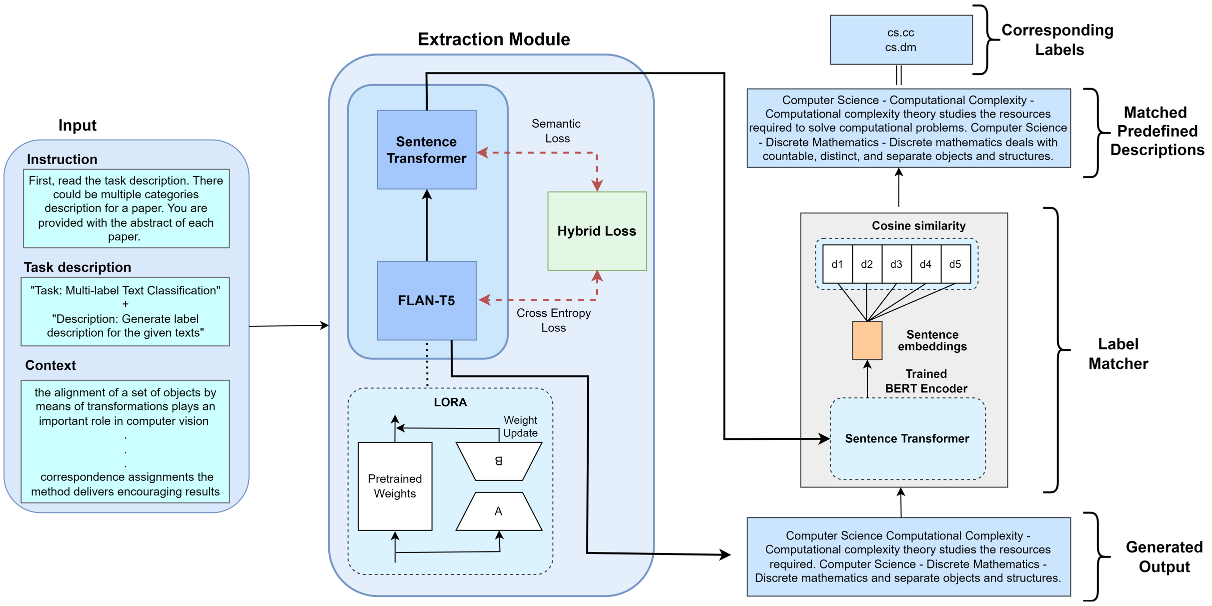

Figure 1: Our proposed framework. Extraction module takes as input a task-specific instruction and the input text to classify. In this module, FLAN-T5 is trained along with a Sentence transformer on a dual objective loss. FLAN-T5-generated label descriptions subsequently flows into the Label Matcher that predicts the final labels for that text using the trained Sentence transformer.

5 Proposed Framework for Multi-label Classification

The proposed framework LAGAMC for multi-label classification (Figure 1), is divided into two stages: a supervised generative phase, and an unsupervised description-to-label matching phase.

Problem Formulation: Given an input sequence $x_{input} = \{x_1, x_2, ..., x_n\}$, the task is to assign text-class labels $Y_k = \{y_1, y_2, ..., y_k\} \subset Y$ to x_{input} (the text to classify) where $Y = \{y_1, y_2, ..., y_p\}$ contains all possible labels of that dataset. We adopt prompt-based learning paradigm, generating text conditioned on a given input prompt.

5.1 Generative Phase

In the first stage, we frame the problem as a generative task, instructing the model to generate label descriptions from a given document using task-specific prompts.

Prompt Construction: We construct a prompt x_{prompt} comprising three components:

Instruction (x_{inst}): provides a brief overview of input (see Table 2 for an example).

Task description (x_{desc}): describes the exact task which needs to be performed. For example in our work when the task the x_{desc} is -

"Task: Multi-label Text Classification
Description: Generate label description for the given texts."

Input Text (x_{input}): This is the input text sequence which in our case can range from an abstract of a paper to a tweet. The x_{prompt} is constructed by concatenation of the Instruction x_{inst}, Task Description x_{desc} and Input Text x_{input}. An example from the SemEval dataset is given in Table 2.

Prompt (x_{prompt})	**Target** (y_{target})
Instruction: First read the task description. There could be multiple categories description for a tweet. **Task**: Multi-label Text Classification **Description**: Generate label description for the given texts. *It's hot as shit and its fogging up my glasses.*	**Anger**, which can also encompass annoyance and rage, is a powerful emotion that arises when one feels slighted or wronged. **Disgust**, which can involve disinterest, dislike, and even loathing, is the strong aversion or revulsion towards something unpleasant or offensive.

Table 2: Example of prompt (instruction and input text) and target (the label descriptions, separated by a full-stop) from the SemEval dataset.

5.2 Response Construction

The proposed generative model is expected to generate a textual response y_{target} which is a concatenation of pre-defined label description of the true labels of the corresponding text. So, if the expected output has k labels = $\{y_1, y_2, ..., y_k\}$ then y_{target} = $\{ y'_1 . y'_2 .... y'_k \}$ where y'_i denotes the pre-defined label description for the i^{th} label (concatenated and separated using a stop). Example of y_{target} can be seen in the *Target* column of Table 2. In this generative phase, we formulate x_{prompt}, y_{target} for each data point in training dataset. We provide this x_{prompt} as input with target as y_{target} to our model.

5.3 Hybrid Loss

In text generation tasks, models like FLAN-T5 are trained with cross-entropy (CE) loss. Cross-entropy loss operates at a token level, meaning it only rewards exact matches at each position in the sequence. The primary limitation arises from its inability to account for semantically equivalent sentences that use different tokens. Hence, we incorporate a semantic similarity based term in the loss function while training the generative model, which prevents the model from overfitting to exact matches. Using such a hybrid loss helps bring the embeddings of the outputs of the generative model and the embeddings of the predefined label description closer in the representation space, making it easier to map the generated outputs with final labels. We define the hybrid loss function as follows:

$$\mathcal{L}_{\text{hybrid}} = \lambda \cdot \mathcal{L}_{\text{CE}} + (1 - \lambda) \cdot \mathcal{L}_{\text{semantic}} \quad (1)$$

where: $\mathcal{L}_{\text{CE}} = -\sum_{t=1}^{T} y_t \log(\hat{y}_t)$ represents the traditional cross-entropy loss, which is computed at the token level. Here, y_t is the ground-truth token at position t, and $\hat{y}_t$ is the predicted probability for the token at the same position.

$\mathcal{L}_{\text{semantic}} = 1 - \text{CosSim}(v_{\text{gen}}, v_{\text{target}})$ is the semantic similarity loss, where v_{gen} and v_{target} represent the embeddings produced by the sentence transformer for the generated output from generative model and the target y_{target}, respectively. $CosSim$ denotes the Cosine similarity between the generated and target embeddings. The *sentence transformer is allowed to train* and adapt during the learning process. λ is a learnable parameter that dynamically adjusts the balance between cross-entropy loss and semantic similarity loss.

5.4 Label Matching Phase

During inference, we employ a *Label Matcher* module to assign labels based on similarity between generated and predefined descriptions. We utilize the *trained sentence transformer from the generative phase* to obtain embeddings for both the generated sentences $\{gendesc_1, gendesc_2, \ldots, gendesc_k\}$ and the predefined label descriptions. For each generated sentence $gendesc_i$, we compute its cosine similarity with all label embeddings and select the label with the highest similarity as the final prediction $predLabel_i$. This approach ensures robust matching, even when the generated descriptions deviate from the predefined labels.

5.5 Generative Models Explored

To evaluate the viability of a generative approach for this task, we conduct a comprehensive assessment of multiple generative models, varying in size and training strategies. First, we fine-tune T5-Base (220M parameters) and T5-Large (780M parameters), to generate target descriptions. Next, we fine-tune FLAN-T5 Large (Longpre et al., 2023; Chung et al., 2022), which benefits from extensive pre-training on over 1.8K instruction-based tasks. For efficient fine-tuning, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2021) for all the generative models updating just 0.08% of model parameters. Details on trainable parameters for our models and baselines are provided in Table 3.

6 Experimental Setup

For all our model variants (performed on NVIDIA A100 80G GPUs), we obtain the pre-trained checkpoints from the *Huggingface* Library[7]. For training the models with LoRA, the *rank* for the trainable decomposition matrices was set to 2. FLAN-T5-Large model is instruction-tuned for 20 epochs, with batch-size of 8 and with an lr of $2e-4$ with LoRA (training time: 56 minutes/epoch, inference time: 2 minutes/sample). These hyperparameters were selected based on the best Macro-F1 results on the validation set. The input length was set by the average number of input tokens per dataset, while the output length was based on the average label description length for each dataset.

7 Results and Discussion

We report the results of our proposed generative model variants and various baselines in Table 3 for all the datasets. We first finetune pretrained transformers such as BERT, XLNet, RoBERTa with a projection layer for the task of MLTC. Next we compare with baselines designed specifically for the task of MLTC such as GalaXC, DeepXML, Renee, etc. We observe that among the baselines the models designed specifically for MLTC task outperform the finetuned transformers. To check the robustness of our proposed generative framework,

[7] https://huggingface.co/

	CAVES		SemEval		Reuters		AAPD		PubMed		#TP(M)
Models	**Mi-F1**	**M-F1**	**Mi-F1**	**M-F1**	**Mi-F1**	**M-F1**	**Mi-F1**	**M-F1**	**Mi-F1**	**M-F1**	
Baselines											
BERT	70.36	65.29	70.70	56.30	87.73	34.98	71.30	55.90	85.05	70.99	110
XLNet	71.61	63.83	58.01	35.31	88.54	51.99	70.07	58.39	85.33	70.81	110
RoBERTa	71.34	63.82	59.82	40.55	88.27	42.63	69.14	54.88	85.19	70.54	125
TextCNN	55.48	39.64	54.55	39.51	81.89	33.96	67.71	49.85	82.62	66.4	3.88
TextRNN	57.70	42.17	52.42	37.94	81.72	33.26	69.28	52.27	83.11	67.72	3.86
StarTransformer	53.86	35.98	51.42	38.96	80.22	36.39	68.22	49.36	82.35	67.35	3.84
AttentiveConvNet	54.22	38.15	51.61	37.21	79.77	31.86	68.11	49.21	82.65	66.33	3.91
ChatGPT	57.22	44.47	41.61	27.21	69.77	35.86	48.11	29.21	42.65	26.33	-
AttentionXML	67.12	52.83	61.55	48.32	78.43	43.57	69.01	56.28	84.39	69.11	112
GalaXC	69.84	56.12	62.34	50.79	81.23	46.39	70.65	58.17	85.91	71.34	41
SiameseXML	72.16	59.89	65.42	52.84	84.22	48.99	72.48	60.43	86.42	72.98	115
DEXA	74.81	62.35	66.57	54.32	86.07	51.12	74.21	63.12	87.25	74.22	134
DeepXML	77.53	65.84	68.76	55.98	88.52	53.79	75.63	64.58	88.41	75.96	161
Renee	79.46	67.93	71.18	58.43	90.29	66.21	77.82	66.04	89.74	78.12	82
Proposed											
T5-Base	86.84	71.22	83.12	70.13	90.89	69.23	82.25	71.22	89.87	78.56	22.32
T5-Large	88.33	81.89	84.35	72.23	91.12	71.32	84.23	71.59	89.90	79.35	22.68
LAGAMC	**92.46**	**89.11**	**87.81**	**78.06**	**96.48**	**80.85**	**95.64**	**88.43**	**89.93**	**80.81**	22.69

Table 3: Performance evaluation based on Micro F1 and Macro F1 scores across multiple datasets. The best performance is highlighted in bold, and the strongest baseline result is underlined. The last column TP (M) indicate approx. no of trainable parameters in million.

we try out with different base models such as T5-Base, T5-Large. Our devised generative baselines outperform the best baselines across all datasets. We also report ChatGPT's performance by providing the same prompt as provided to LAGAMC.

LAGAMC on an average improves by **13.94%** in Micro-F1 and **24.85%** in Macro-F1 when compared with the best baseline for the given dataset. The best performance boost is seen for SemEval [8] and the second best for the CAVES dataset. The performance of LAGAMC compared to the respective SOTA models in domains ranging from tweet sentiment to medical domain dataset, academic text showcases its adaptability and versatility.

7.1 Parameter Efficiency

The last column of Table 3 compares the number of trainable parameters across different baselines, including the generative models we trained. Our best-performing model, LAGAMC along with our generative baselines, demonstrates parameter efficiency by having significantly fewer trainable parameters compared to most of the closest competing baselines.

[8] Public leaderboard available at https://paperswithcode.com/sota/emotion-classification-on-semeval-2018-task

7.2 Utility of Label Descriptions

To understand the importance of label descriptions, we perform an experiment where we set atomic labels instead of their descriptions, as the target. Accordingly, the *Label Matcher* module now compares the embeddings of the generated and ground truth labels (and not descriptions). From the Table 5, we observe a average drop of **36.04%** in Macro-F1.

7.3 Utility of Semantic Loss

We assess the significance of semantic loss by comparing the performance of our proposed hybrid loss function with that of the standard cross-entropy loss. The results, summarized in Table 5, show a drop of **3.96%** and **7.33%** in Micro-F1 and Macro-F1, respectively, when using only cross-entropy loss.

8 Analysis

We now present different analyses and ablations of LAGAMC.

8.1 Zero Shot Capability

To evaluate the zero shot capability of our proposed model, we constructed a test dataset with labels not seen during training. Specifically, 4-5 labels from each dataset were randomly selected as unseen labels, appearing only in test instances. We then trained the model on the modified dataset to assess its ability to predict unseen labels. As shown in Figure 2, the average Macro-F1 score across all datasets was 83.45 with full training, compared to 70.61 in the zero-shot setting—demonstrating strong performance in the more challenging zero-shot scenario. This reduces the need for retraining and also accelerates real-world deployment.

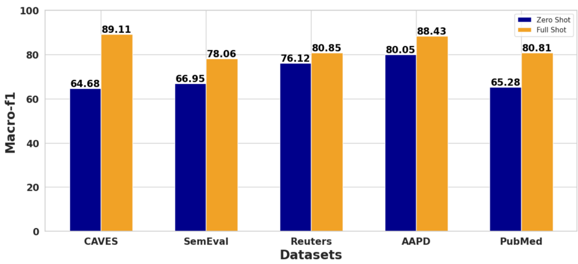

Figure 2: Zero-shot performance of LAGAMC , achieving an average macro-F1 score of 70.61.

8.2 Performance on Least Frequent Labels

We evaluate the model's performance on the least frequent labels as this is critical for real-world applications, where rare labels may represent significant events. We identified the least frequent 15% of labels from each dataset's training set and computed the Macro F1-score on test samples where the ground truth labels were part of this rare label set. As illustrated in Figure 3, our model demonstrates superior performance over the closest baseline across all datasets, with an average improvement of **22%** in Macro-f1.

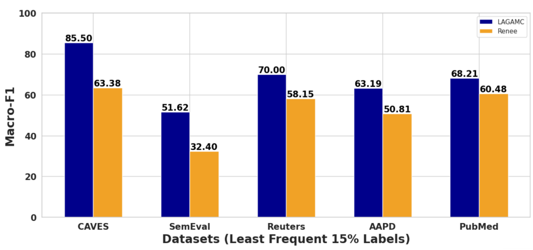

Figure 3: Comparison of Model Performance on Least Frequent Labels: Our proposed model demonstrates superior performance compared to closest baseline.

8.3 Evaluation of Recent LLM's

We also evaluated recent LLMs, such as Llama-2-7b (Touvron et al., 2023), Mistral-7B (Jiang et al., 2023) for multi-label classification. The initial results shown in Table 4 were promising, as these models outperformed all baseline methods (except Micro-F1 on PubMed which is close to the best baselines). However, their accuracy was lower than our proposed LAGAMC, which uses FLAN-T5. Due to limited GPU resources, we could not fully fine-tune these models or conduct an extensive hyperparameter search for LoRA fine-tuning, which likely contributed to the lower performance. We also expect that increasing the context length could improve results. These findings indicate that our pipeline is effective and can be applied to other LLMs for multi-label classification. We also evaluated our pipeline with Llama-3.1-7B (Grattafiori et al., 2024) and observed an improvement of nearly 1% in Micro-F1 and 2% in Macro-F1.

Dataset	Model	Mi-F1 / M-F1
CAVES	Ours	92.46 / 89.11
	Ours with threshold	91.46 / 88.55
	Llama-2-7B with threshold	90.97 / 87.13
	Llama-2-7B w/o threshold	89.67 / 83.65
	Mistral-7B with threshold	90.17 / 86.52
	Mistral-7B w/o threshold	88.59 / 82.25
SemEval	Ours	87.81 / 78.06
	Ours with threshold	86.18 / 77.60
	Llama-2-7B with threshold	86.54 / 77.11
	Llama-2-7B w/o threshold	84.67 / 75.65
	Mistral-7B with threshold	85.54 / 76.25
	Mistral-7B w/o threshold	84.45 / 74.45
Reuters	Ours	96.48 / 80.85
	Ours with threshold	94.48 / 76.15
	Llama-2-7B with threshold	93.62 / 74.65
	Llama-2-7B w/o threshold	92.18 / 72.88
	Mistral-7B with threshold	95.17 / 76.43
	Mistral-7B w/o threshold	93.55 / 74.15
AAPD	Ours	95.64 / 88.43
	Ours with threshold	94.46 / 86.73
	Llama-2-7B with threshold	93.46 / 86.11
	Llama-2-7B w/o threshold	89.12 / 75.97
	Mistral-7B with threshold	92.76 / 86.07
	Mistral-7B w/o threshold	87.57 / 74.15
PubMed	Ours	89.93 / 80.81
	Ours with threshold	89.22 / 78.91
	Llama-2-7B with threshold	89.92 / 79.71
	Llama-2-7B w/o threshold	86.75 / 77.45
	Mistral-7B with threshold	89.67 / 79.55
	Mistral-7B w/o threshold	87.65 / 78.35

Table 4: Performance comparison (Mi-F1 / M-F1) on multiple datasets using different LLMs.

8.4 Robustness of the Model

To assess the robustness of our approach, we created a unified dataset by randomly selecting 500 training samples and 100 test samples from each dataset. The average Macro-F1 score was 83.45 across all datasets, compared to 78.38 for the mixed dataset. Despite predicting from 181 ground truth labels (sum of Labels column in Table 1), our model shows only a 5.07% drop in performance, while the closest baseline declines by 14.3% .

Dataset	Models	Mi-F1 / M-F1
CAVES	Ours	92.46 / 89.11
	w/o Semantic loss	89.67 / 85.25
	w/o Label Description	67.63 / 62.98
SemEval	Ours	87.81 / 78.06
	w/o Semantic loss	85.53 / 74.79
	w/o Label Description	64.24 / 54.35
Reuters	Ours	96.48 / 80.85
	w/o Semantic loss	94.97 / 78.12
	w/o Label Description	68.24 / 43.12
AAPD	Ours	95.64 / 88.43
	w/o Semantic loss	86.94 / 73.13
	w/o Label Description	65.27 / 53.28
PubMed	Ours	89.93 / 80.81
	w/o Semantic loss	86.77 / 74.12
	w/o Label Description	67.29 / 53.21

Table 5: Performance comparison with, without label descriptions and without Semantic loss across datasets. Results highlight the importance of label descriptions and Semantic loss for multi-label classification.

Descriptions Length vs Performance: We evaluate our model's performance based on the length of concatenated label descriptions. To do this, we group label descriptions into buckets with an equal number of test samples. Longer descriptions correspond to more ground truth labels, increasing prediction complexity. Figures 4 and 5 show a slight performance drop for very long descriptions, a trend consistent across datasets.

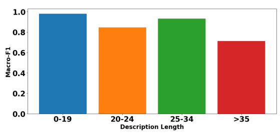

Figure 4: CAVES

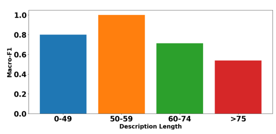

Figure 5: SemEval

Figure 6: Impact of label description length

Actual vs Predicted No of Labels: For each dataset, we analyze the number of samples with a given number of labels and compare it to the number of samples predicted to have the same number of labels (Table 6). We have analyzed upto five label counts. The model tends to give single-label predictions for the SemEval and Caves datasets.

8.5 Analysis of Label Matcher Module

We examine the computational efficiency of the module and propose a threshold-based approach to prevent label assignments from hallucinated text.

Computation Efficiency: The Label Matcher module (Section 5.4) assigns labels by computing cosine similarity between sentence and label embeddings. Using NumPy's (Harris et al., 2020) matrix operations for parallel computation significantly improves efficiency. For example, with 10,000 sentences and 1,000 labels using 1,024-dimensional embeddings, the matrix-based approach completes in 0.089s, compared to 0.354s with the sequential method. In the worst-case with all 1,000 labels present in a single instance, inference takes just 0.007s (matrix-based) versus 0.043s (sequential).

Hallucination in Predictions: During label matching, each output sentence is assigned the nearest label based on cosine similarity. However, LLM-generated text may include hallucinated sentences, leading to incorrect predictions. To mitigate this, we enforce a minimum similarity threshold, ensuring a sentence is assigned a label only if its highest similarity score exceeds a set value. Our analysis finds 0.4 threshold to be optimal. As shown in Table 4, this slightly reduces performance for our model (FLAN-T5-based) but improves results for larger LLMs like Llama-2 and Mistral, likely due to their tendency to generate longer outputs.

Labels	Caves	SemEval	Reuters	AAPD	PUBMED
1	(1386, 1562)	(288, 456)	(2592, 2583)	(-, -)	(-, -)
2	(579, 369)	(1486, 1367)	(279, 308)	(642, 690)	(50, 105)
3	(12, 46)	(1078, 1055)	(86, 63)	(264, 225)	(376, 535)
4	(-, -)	(395, 316)	(32, 32)	(69, 62)	(1438, 1503)
5	(-, -)	(11, 60)	(17, 15)	(23, 21)	(2200, 2404)

Table 6: Comparison of actual and predicted sample counts based on number of labels. Each cell (x, y) indicates the number of actual samples (x) and the number of predicted samples (y) for a specific label count.

Model	CAVES (M-F1)	SemEval (M-F1)	Reuters (M-F1)	AAPD (M-F1)	PubMed (M-F1)
Ours	89.11	78.06	80.85	88.43	80.81
w S-BERT-L12	82.80	74.30	58.60	71.40	74.10
w ST5-xxl	83.20	74.70	56.70	71.40	75.00
w/o Instruction	81.30	74.80	56.40	71.60	74.30

Table 7: Ablation study results: M-F1 scores for LAGAMC and its variations.

8.6 Ablation study of model components

We conduct ablations on our best model to assess module significance. For *Label Matcher*, replacing fine-tuned Sentence-BERT-Transformer with Sentence-T5-xxl or Sentence-BERT-L12 (Reimers and Gurevych, 2019) lowers performance (Table 7). Similarly, instruction-tuning FLAN-T5-Large *without task-specific instructions* results in a performance drop across all datasets, highlighting the importance of instruction alignment.

8.7 Effect of Label Descriptions in Existing Models

We conducted additional experiments by integrating label descriptions into two strong baselines: BERT and DeepXML. For the BERT-based multi-label classification baseline, we adopted a joint encoding strategy where both the input text and the label descriptions are encoded using a shared BERT encoder. This allows the model to learn interactions between label semantics and text representations.

For DeepXML, which supports metadata incorporation, we introduced label descriptions as auxiliary features. Additionally, during the clustering phase, we replaced label names with their corresponding descriptions to influence label partitioning based on semantic content.

We evaluated these modified models on the CAVES and SemEval datasets. The results are presented in Table 8. We observe that incorporating label descriptions provides consistent but modest performance improvements over the original versions of BERT and DeepXML. However, our proposed generative approach with the label matcher module achieves significantly better performance, demonstrating the advantage of a design that integrates label semantics into the prediction process.

8.8 Error Analysis

To characterize the errors committed by our model, we check when our model predicts wrong label or provides a subset of ground truth labels. We notice that our model sometime struggles against complex inputs. The first example stated in Table 9 is about a news related to a company dealing with gold but the news excerpt is regarding acquisition of it. This confuses our model because even though the word gold is mentioned multiple times, the main subject of the news is regarding the acquisition rather than about gold commodities.

Sometime due to limitations in input context, our model may not predict all corresponding labels accurately. Instead, it tends to predict a subset of labels. Our model outperforms the best baseline by accurately distinguishing correlated labels, such as ‘joy’, ‘love’, and ‘optimism’, which frequently co-occur. While the baseline model misclassified approximately 33% of samples predicting as, ‘joy love optimism’, our model correctly predicts for all such samples. One such example is shown in last row of Table 9.

Model	CAVES Mi-F1 / M-F1	SemEval Mi-F1 / M-F1
BERT	70.36 / 65.29	70.70 / 56.30
BERT + Label Desc.	73.00 / 67.50	72.20 / 58.00
DeepXML	77.53 / 65.84	68.76 / 55.98
DeepXML + Label Desc.	79.50 / 67.20	70.12 / 57.30

Table 8: Performance of baselines with and without label descriptions.

Dataset	Abstract	Ground Truth	Ours	Renee
Reuters	CRA SOLD FORREST GOLD FOR 76 MLN DLRS... It also owns an undeveloped gold project.	acq	acq, gold	acq, gold
CAVES	The covid vaccine is not a vaccine... the next round of manufactured flu.	conspiracy ineffective side-effect	conspiracy	side-effect
SemEval	I used to make the peanut butter energy balls all the time. My famjam loved them!	joy, love	joy, love	joy, love, optimism

Table 9: Examples showing errors in predictions by our model and the closest baseline (Renee).

9 Conclusion

In this work, we propose a parameter-efficient generative approach equipped with a dual loss objective to tackle the challenging problem of multilabel classification. Our method introduces a novel and domain-agnostic framework that is flexible enough to be adapted across various applications. By leveraging both generative modeling and discriminative supervision, the approach effectively captures label correlations and enhances prediction robustness. Through extensive experiments, we compare our method against several state-of-the-art models and strong baseline systems specifically designed for the task. Our results show that LAGAMC achieves significant performance gains, demonstrating its superiority across multiple evaluation metrics. Furthermore, we conduct detailed ablation studies and empirical analyses to validate the contribution of each component within the framework.

10 Limitations

A limitation of our proposed framework is that the approach relies on the availability of label descriptions, which may not always be readily accessible and would need to be generated when absent. Additionally it has not been tested on extreme multi-label classification datasets.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: Bert for document classification.

Iqra Ameer, Necva Bölücü, Muhammad Hammad Fahim Siddiqui, Burcu Can, Grigori Sidorov, and Alexander Gelbukh. 2023. Multi-label emotion classification in texts using transfer learning. *Expert Systems with Applications*, 213:118534.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. 2014. Senticnet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. volume 2.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Guibin Chen, Deheng Ye, Zhenchang Xing, Jieshan Chen, and Erik Cambria. 2017. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2377–2383.

Xiaolong Chen, Jieren Cheng, Jingxin Liu, Wenghang Xu, Shuai Hua, Zhu Tang, and Victor S Sheng. 2022. A survey of multi-label text classification based on deep learning. In *International Conference on Adaptive and Intelligent Systems*, pages 443–456. Springer.

Hyung Won Chung et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Kunal Dahiya, Ananye Agarwal, Deepak Saini, K Gururaj, Jian Jiao, Amit Singh, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021a. Siamesexml: Siamese networks meet extreme classifiers with 100m labels. In *International conference on machine learning*, pages 2330–2340. PMLR.

Kunal Dahiya, Deepak Saini, Anshul Mittal, Ankush Shaw, Kushal Dave, Akshay Soni, Himanshu Jain, Sumeet Agarwal, and Manik Varma. 2021b. Deepxml: A deep extreme multi-label learning framework applied to short text documents. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 31–39.

Kunal Dahiya, Sachin Yadav, Sushant Sondhi, Deepak Saini, Sonu Mehta, Jian Jiao, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2023. Deep encoders with auxiliary parameters for extreme classification. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 358–367.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Siddharth Gopal and Yiming Yang. 2010. Multilabel classification with meta-level features. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick

Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2022. Star-transformer.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Philip J. Hayes and Steven P. Weinstein. 1990. Construe/tis: A system for content-based indexing of a database of news stories. In *Conference on Innovative Applications of Artificial Intelligence*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Yi Huang, Buse Giledereli, Abdullatif Köksal, Arzucan Özgür, and Elif Ozkirimli. 2021. Balancing methods for multi-label text classification with long-tailed class distribution.

Vidit Jain, Jatin Prakash, Deepak Saini, Jian Jiao, Ramachandran Ramjee, and Manik Varma. 2023. Renee: End-to-end training of extreme classification models. *Proceedings of Machine Learning and Systems*, 5.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Siddhant Kharbanda, Atmadeep Banerjee, Devaansh Gupta, Akash Palrecha, and Rohit Babbar. 2024. Inceptionxml: A lightweight framework with synchronized negative sampling for short text extreme classification.

Subhendu Khatuya, Rajdeep Mukherjee, Akash Ghosh, Manjunath Hegde, Koustuv Dasgupta, Niloy Ganguly, Saptarshi Ghosh, and Pawan Goyal. 2024. Parameter-efficient instruction tuning of large language models for extreme financial numeral labelling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7391–7403, Mexico City, Mexico. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Shayne Longpre et al. 2023. The flan collection: Designing data and methods for effective instruction tuning.

Long Ma, Zeye Sun, Jiawei Jiang, and Xuan Li. 2023. PAI at SemEval-2023 task 4: A general multi-label classification system with class-balanced loss function and ensemble module. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 256–261, Toronto, Canada. Association for Computational Linguistics.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models.

Soham Poddar, Azlaan Mustafa Samad, Rajdeep Mukherjee, Niloy Ganguly, and Saptarshi Ghosh. 2022. Caves: A dataset to facilitate explainable classification and summarization of concerns towards covid vaccines. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Colin Raffel et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Deepak Saini, Arnav Kumar Jain, Kushal Dave, Jian Jiao, Amit Singh, Ruofei Zhang, and Manik Varma. 2021. Galaxc: Graph neural networks with labelwise attention for extreme classification. In *Proceedings of the Web Conference 2021*, pages 3733–3744.

Robert Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning - ML*, 39:135–168.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-specific document representation for multi-label text classification. In *Proceedings of the 2019 conference in EMNLP-IJCNLP*, pages 466–475.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. Sgm: Sequence generation model for multi-label classification.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of NAACL: human language technologies*, pages 1480–1489.

Hui Ye, Rajshekhar Sunderraman, and Shihao Ji. 2024. Matchxml: An efficient text-label matching framework for extreme multi-label text classification.

Wenpeng Yin and Hinrich Schütze. 2018. Attentive convolution: Equipping cnns with rnn-style attention mechanisms. *Transactions of the Association for Computational Linguistics*, 6:687–702.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *Advances in neural information processing systems*, 32.