From Nearest Neighbors to LLMs: A Hybrid Zero-Shot Approach to Multi-Label Classification

Konstantinos Bougiatiotis[1,2,*], Georgios Paliouras[1]

[1]Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", Athens, Greece
[2]Department of Informatics and Telecommunications, National and Kapodistrian University, Athens, Greece

Abstract

This paper presents our approach for the FoRC4CL shared task of NSLP 2025, which aims to classify computational linguistics publications into a predefined three-tiered taxonomy of 181 categories. We explore various strategies, including k-Nearest Neighbors (k-NN) on sentence embeddings, graph neural networks, and incorporating Large Language Models (LLMs) for label refinement in a zero-shot scenario. Our best-performing approach involves using a k-NN model to fetch candidate labels, followed by an LLM model for prediction. We achieve a macro F1-score of 0.661 on the public test leaderboard, surpassing the nearest competitor by 20%. Our results demonstrate that a retrieval-enhanced LLM-based zero-shot approach, with a posteriori enforced hierarchical consistency, is a feasible solution for this task.

Keywords

Multi-label classification, Hierarchical classification, Large Language Models, Graph Neural Networks

1. Introduction

The process of automatically assigning relevant subject headings or labels to academic papers in libraries is known as "Automatic Subject Indexing". In recent years, to assess document content and identify relevant subject categories, this procedure makes use of Machine Learning and Natural Language Processing (NLP). This is the theme of the Field of Research Classification for Computational Linguistics (FoRC4CL) shared task at NSLP2025. The goal is to classify computational linguistics publications into a predefined three-tiered taxonomy of 181 categories. Each paper is assigned multiple subject labels, making this a hierarchical multi-label classification problem. The dataset is highly imbalanced, with some categories appearing far more frequently than others. Furthermore, the hierarchical nature of the taxonomy introduces dependencies between labels, requiring models to ensure consistency across different levels.

Our approach explores multiple lines of research, including:

1. k-Nearest neighbors (k-NN) with Sentence Embeddings: Retrieving candidate labels based on semantic similarity of texts.
2. Graph Neural Networks for Text Classification: Representing papers as nodes in a k-NN distance graph and modeling the task as multi-label node classification.
3. Zero-Shot Classification using Large Language Models (LLMs): Using an LLM to select the labels while enforcing hierarchy constraints based on Nearest Neighbors models.

Our best-performing approach involves a two-stage pipeline. First, candidate labels are retrieved using a k-NN index based on sentence embeddings, and second, an LLM refines the predictions while ensuring hierarchical validity. Compared to traditional approaches, this method improves classification performance by leveraging both local similarity-based retrieval and the global reasoning capabilities of LLMs.

2nd International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2025), co-located with ESWC 2025, June 01–02, 2025, Portorož, Slovenia

*Corresponding author.
kbogas@di.uoa.gr (K. Bougiatiotis)
0000-0002-1910-2758 (K. Bougiatiotis); 0000-0001-9629-2367 (G. Paliouras)

The rest of this paper is structured as follows: Section 2 discusses related work, and Section 3 describes the dataset used in the shared task. Section 4 outlines the models created for this task, followed by the evaluation results and discussion in Section 5. Finally, we conclude with future directions in Section 6[1].

2. Related Work

Hierarchical multi-label text classification (HMTC) assigns documents to multiple categories arranged in a hierarchy, posing unique challenges due to label dependencies and the hierarchical structure. Traditional methods, such as tree-based approaches like BONSAI [1] and PARABEL [2], have been widely used to address the extreme multi-label classification (XMLC) problem. Building upon this foundation, models like X-Transformer [3] introduce innovations such as X-Linear and XR-Transformers, which recursively fine-tune pre-trained transformers for improved performance. Specifically for this task, the weak X-Transformer [4] was introduced in the previous FoRC4CL shared task.

Other recent advancements have explored the integration of Graph Neural Networks (GNNs) to capture complex relationships in HMTC tasks. For instance, the Hierarchical Graph-Based Label Propagation (HGBL) model constructs a heterogeneous graph of the entire corpus and employs Graph Convolutional Networks (GCNs) to learn document representations, effectively capturing word order and semantic relationships [5]. Similarly, the Hierarchical Multi-label Text Classification with Horizontal and Vertical Category Correlations (HV-HMC) model utilizes a loosely coupled GCN to extract representations for words and documents, emphasizing both horizontal and vertical category correlations [6].

Large Language Models (LLMs) have also been applied to text classification tasks. The Clue And Reasoning Prompting (CARP) method, for example, enhances LLMs' reasoning abilities to address complex linguistic phenomena in text classification [7]. In the healthcare domain, ensemble learning approaches combining LLMs have shown promise in handling the nuanced nature of multi-label text classification [8]. Despite these advancements, challenges remain in achieving consistent performance across all levels of the hierarchy and effectively integrating hierarchical dependencies into model architectures. Our work aims to build upon these approaches by exploring the synergy between retrieval methods and LLMs.

3. Shared Task Data

The dataset used in this task consists of computational linguistics publications, structured across three hierarchical levels of subject labels. This year's shared task data comprised 1,050 training, 250 evaluation, and 250 test samples. Each record contains multiple metadata fields, including the ACL Anthology ID, title, abstract, author(s), URL to the full text, publisher, publication year and month, proceedings title, DOI, and venue. This year, each sample was also enriched with the corresponding publication's full text.

Table 1
Statistics of the labels partitioned per hierarchy level. The last column corresponds to the average number of children labels for non-leaf node labels in each level.

Hierarchy Level	# Unique	Mean #	Min #	Max #	Mean # Children
Level 1	46	3.2	3.0	1.0	3.0
Level 2	109	1.8	2.0	0.0	1.0
Level 3	26	0.4	0.0	0.0	-

Each publication is assigned multiple hierarchical labels (among 181 available) across three levels,

[1]The code for reproduction and further experimentation is available at https://github.com/kbogas/DICE-FoRC25.

ranging from general to fine-grained categories, following a predefined taxonomy[2]. These annotations were curated based on subject extraction techniques and aligned with topics derived from multiple paper sources. The reliability of these annotations was evaluated using inter-annotator agreement scores [9]. Details on the characteristics of the labels per hierarchy level can be seen in Table 1. For example, we notice that the third-level labels are much fewer than the second-level ones and much sparser than the first-level ones, hinting at the task's difficulty.

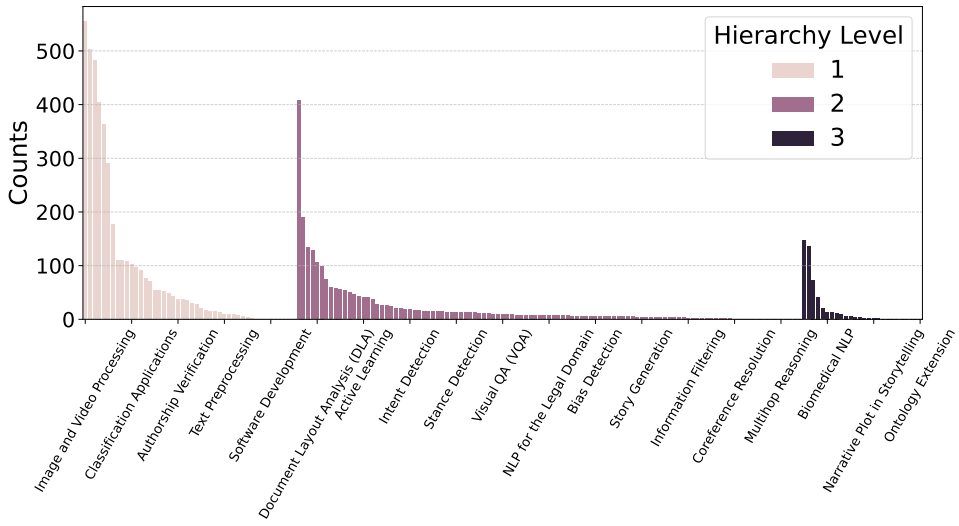

Figure 1: Label counts, across hierarchy levels, in the training set. The further down the taxonomy tree, the sparser the label occurrences become.

A significant challenge of the dataset is the severe imbalance in label distribution, as illustrated in Figure 1. Certain labels appear infrequently, and some labels in the validation and test sets are absent from the training data. In addition to the primary dataset, this year's frozen splits are accompanied by weakly supervised data of approximately 41,000 ACL publications. In our current experiments, these weakly supervised data were not utilized.

4. Methodology

In this section, we provide details on the three main lines of research we worked on.

4.1. Sentence Transformer and Nearest neighbors

Building upon prior research using BERT-based and other transformer-based models for this task [4], our initial objective was to establish a strong baseline with minimal computational overhead. To achieve this, we used Sentence Transformers [10], which have demonstrated effectiveness as off-the-shelf embedding models in multiple domains.

Our first approach combined Sentence Transformer embeddings with a k-Nearest neighbors (k-NN) classifier on top (denoted ST + k-NN). Specifically, we computed embeddings for all training samples and, at inference time, embedded the query document and retrieved its closest training samples based on cosine similarity. The final labels of the query sample were determined using a distance-weighted voting scheme, where labels from the nearest neighbors were aggregated proportionally to the samples' similarity scores. This baseline served as a rapid prototyping framework, enabling efficient experimentation and metadata selection. Specifically, after a small-scale grid search, we adopted a setup where each training sample was represented by its title, abstract, and concatenated labels. The query samples were embedded using only the title and abstract.

[2]See the full taxonomy tree in https://huggingface.co/spaces/DFKI-SLT/Taxonomy4CL.

It is interesting to note that we can achieve high recall levels despite this baseline's simplicity. This can be observed in Figure 2, which illustrates the recall across different hierarchy levels as the number of nearest neighbors increases. Even with a relatively small neighborhood size (e.g., $k = 20$), we can achieve approximately 95% recall for Level 1 labels and around 90% on Levels 2 and 3.

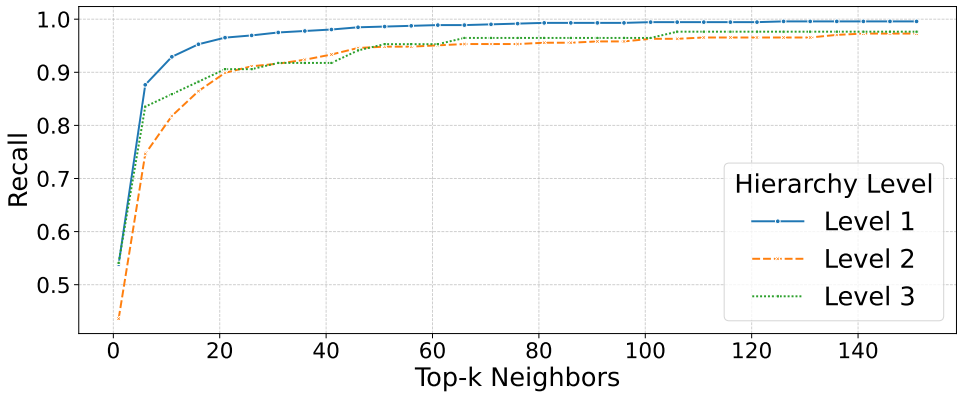

Figure 2: Average recall across all labels for different number of neighbors over the different hierarchy levels. Note that we simply check for each sample's labels' existence in the label set of their neighbors; no classification is done here.

In terms of the hyperparameters for this method, we use a fixed top-$k = 19$, based on the elbow of the chart in Figure 2, cosine similarity as the embedding distance metric, and the default *all-MiniLM-L6-v2* model for encoding each text. Regarding the final classification, we also have a voting threshold fixed at 0.3, denoting that if approximately[3] $19 \times 0.3 \approx 6$ of the neighbors exhibit a specific label, we mark it as a valid prediction for the query sample at hand.

4.2. Graph Neural Networks from k-NN Graphs

Given the strong performance of the baseline model, we sought to further exploit the structure of the nearest-neighbor relationships by creating an explicit graph structure. Inspired by prior work demonstrating the effectiveness of graph-based approaches for text classification [11], we explored Graph Neural Networks (GNNs) as a second line of research.

We first construct a k-NN graph in which each document represents a node, and edges are formed based on their nearest-neighbor connections as explained before. The sentence transformer embeddings serve as node features, and label prediction is formulated as a multi-label node classification problem.

We experimented with various GNN architectures, including Graph Convolutional Networks (GCNs) [12] and Graph Attention Networks (GATs) [13], as well as different numbers of layers and hyper-parameter settings. The best-performing model was a two-layer GAT, effectively capturing label dependencies within the nearest-neighbor graph. Surprisingly, this approach did not improve classification performance over the ST+k-NN baseline.

4.3. Leveraging Large Language Models for Precision

Our final line of experimentation was motivated by the observation that transformer-based retrieval methods, while achieving high recall, suffered from reduced precision and low macro-F1 scores, denoting low performance on infrequent labels. To address this, we introduced a secondary filtering step using Large Language Models (LLMs) in a zero-shot setting without further fine-tuning.

In this setup, given a query sample, we first retrieve the nearest neighbors using our transformer-based approach and extract their corresponding labels. We then construct a natural language prompt containing the title and abstract of the query document, along with the label set of its nearest neighbors.

[3]Because the voting is distance-weighted, train samples near the query sample hold more "voting power" than the ones further away.

The LLM was tasked with refining these candidate labels and creating the final label set. Figure 3 contains a schematic representation of this workflow.

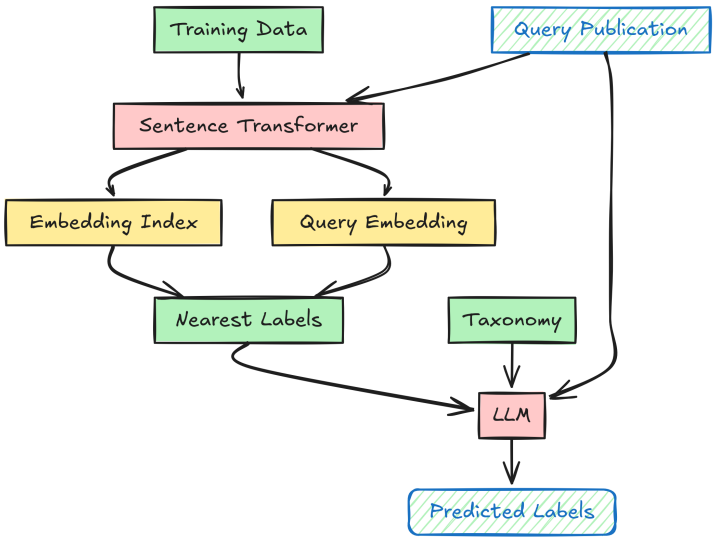

Figure 3: Proposed workflow for improving zero-shot performance of Large Language Models. No training is required for this setup.

We experimented with various open-source LLMs, including the commonly used LLaMA3.1:8B [14], the newer, reasoning-focused and biggest model (in terms of parameters) in our setup DeepSeek-R1:70B [15], and the latest model from Google DeepMind, Gemma3:12B, which has already exhibited great performance in reasoning tasks despite having fewer parameters [16]. We also experimented with different prompt configurations, which mainly encapsulated different ways to present the taxonomy or the possible labels for a specific query sample to the LLMs. The main scenarios tested can be grouped in the following five groups/prompts:

1. Zero Shot-Flat (ZS-Flat): Provide all possible 181 labels to the LLM as a string.
2. Zero Shot-Hier (ZS-Hier): Provide the taxonomy of the labels as a tree-like string.
3. Nearest Labels (NL): Instead of all possible labels, provide only the labels of the query's nearest neighbors, therefore limiting the options of the model.
4. Nearest Labels Count (NL-Count): Provide the labels of the query's nearest neighbors, alongside their occurrence count.
5. Nearest Labels Count + Hier (NL-Count+Hier): The previous configuration in terms of prompt syntax, but we modify the final predictions to adhere to the taxonomy by adding all the "missing ascendant" labels in the predicted label set.

The prompt structure for each of these scenarios is presented in AppendixA. For these models, the underlying k-NN model is the same as before, with the voting threshold set to 0.01 to ensure high-recall in the label set of the nearest neighbors. Moreover, to make sure the LLMs generate valid labels, we validate their output using Pydantic[4] templates, and we fuzzy-match the predicted label strings to the actual labels of the taxonomy. All LLMs are locally hosted using an ollama[5] server.

5. Experiments and Discussion

Table 2 presents the performance of the various models evaluated on the validation set of the shared task. We discuss the key observations and insights derived from these results.

[4]https://docs.pydantic.dev/latest/
[5]https://ollama.com/

The baseline model performs very well, given that it operates in a fully unsupervised manner with fixed hyperparameters. Despite its simplicity, this approach already provides a strong micro F1-score, showcasing its potential on more common labels. In the GNN model, utilizing a 2-layer Graph Attention Network (GAT) layer, we observe a more pronounced drop in macro F1-score compared to micro F1-score, indicating poorer performance on infrequent labels. This aligns with expectations, as the k-NN-based graph construction inherently biases predictions toward more common/homogeneous labels, leading to reduced performance on rare classes.

Table 2
Performance comparison of different models on micro and macro scores on the validation set. Scores for Llama and DeepSeek-R1 are reported using the NL-Count+Hier setup, which was the best one for both of them.

Model	Micro			Macro		
	Prec	Rec	F1-score	Prec	Rec	F1-score
ST + k-NN	0.554	0.575	0.564	0.219	0.201	0.196
GNN (2-layer GAT)	0.500	0.575	0.535	0.139	0.135	0.130
Gemma3 - ZS-Flat	0.414	0.476	0.443	0.294	0.424	0.310
Gemma3 - ZS-Hier	0.312	0.605	0.411	0.184	0.373	0.216
Gemma3 - NL	0.521	0.477	0.498	0.318	0.417	0.330
Gemma3 - NL-Count	0.584	0.571	0.580	0.362	0.425	0.366
Gemma3 - NL-Count + Hier	0.625	**0.688**	**0.655**	**0.371**	**0.453**	**0.389**
Llama 3.1:8B	0.679	0.557	0.612	0.356	0.337	0.321
DeepSeek-R1:70B	**0.688**	0.510	0.584	0.352	0.342	0.322

On the other hand, the first Gemma3-based model, ZS-Flat, significantly improves macro-F1, suggesting better handling of rare labels compared to purely neighborhood-based methods. However, its micro F1-score lags behind the sentence transformer baseline, indicating a lack of sample-wide consistency in label assignments. The hierarchical variant ZS-Hier performs worse than the flat version, suggesting that introducing hierarchical constraints in a zero-shot setting may increase task complexity, thus deteriorating results.

The NL variants show substantial improvement over the zero-shot variants in the micro F1-score. This aligns with expectations, as providing label context from similar examples helps refine predictions without extensive fine-tuning. The NL-Count variant further boosts the micro F1-score, being the first LLM variant to surpass the baseline in this measure. Finally, the best-performing model is the variant which explicitly integrates hierarchical constraints. This yields a significant improvement in the micro F1-score, confirming that previous variants successfully identified fine-grained labels but frequently omitted their higher-level categories. Regarding the other tested LLMS, Llama 3.1:8B delivers results similar to Gemma but slightly worse due to its smaller number of parameters and older architecture. Surprisingly, the DeepSeek-R1:70B model underperforms both Gemma and Llama despite having approximately 9 times the number of parameters of Llama.

Overall, our findings highlight the effectiveness of our initial unsupervised baseline, which performs competitively against many zero-shot models in terms of micro F1-score. This is consistent with our intuition about the dataset's "homophily" where similar samples tend to share labels, making nearest-neighbor-based approaches particularly effective. However, the substantial gains in macro F1-score when using LLMs, even in a zero-shot setting, emphasize the need for better handling of "heterophilous" samples and rare labels.

6. Conclusion

In this study, we explored multiple approaches for multi-label classification of Computational Linguistics scholarly papers as part of the shared task challenge of FoRC4CL, organized by the NSLP 2025 workshop. Our experiments spanned unsupervised retrieval-based methods, graph-based learning, and Large

Language Model zero-shot classification. The best-performing model combined Sentence Transformer embeddings for retrieval, with Gemma3:12B as a zero-shot predictor, leveraging related labels for final predictions. This approach achieved the highest performance among publicly ranked submissions on the task leaderboard. For future work, an interesting direction would be to investigate the complementarity of the different models explored in this study, potentially improving predictions through ensemble methods. Additionally, fine-tuning LLMs specifically for this task could further enhance classification accuracy.

Acknowledgments

This work was funded by the SIMPATHIC project in the context of European Union's Horizon 2020 research and innovation programme under the agreement No 101080249.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 and Grammarly in order to: Drafting content, Grammar and spelling check, Improve writing style. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] J. Singh, M. Varma, Bonsai: diverse and shallow trees for extreme multi-label classification, in: Proceedings of NeurIPS, 2020.

[2] B. Loni, M. Varma, Parabel: Partitioned label trees for extreme classification with application to zero-shot learning, in: Proceedings of ICML, 2018.

[3] W.-C. Chang, H.-F. Yu, I. S. Dhillon, X-transformer: Transformer architectures for extreme multi-label classification, in: Proceedings of AAAI, 2020.

[4] L. R. Bashyam, R. Krestel, Advancing automatic subject indexing: combining weak supervision with extreme multi-label classification, in: Proceedings of the 1st International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2024). Hersonissos, Crete, Greece, volume 27, 2024.

[5] C. Zhang, L. Dai, C. Liu, L. Zhang, Hgbl: A fine granular hierarchical multi-label text classification model, Neural Processing Letters 57 (2024) 1. doi:10.1007/s11063-024-11713-x.

[6] L. Xu, S. Teng, R. Zhao, J. Guo, C. Xiao, D. Jiang, B. Ren, Hierarchical multi-label text classification with horizontal and vertical category correlations, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 2459–2468. doi:10.18653/v1/2021.emnlp-main.190.

[7] X. Sun, X. Li, J. Li, F. Wu, S. Guo, T. Zhang, G. Wang, Text classification via large language models, 2023. URL: https://arxiv.org/abs/2305.08377. arXiv:2305.08377.

[8] H. Sakai, S. S. Lam, Quad-llm-mltc: Large language models ensemble learning for healthcare text multi-label classification, 2025. URL: https://arxiv.org/abs/2502.14189. arXiv:2502.14189.

[9] R. Abu Ahmad, E. Borisova, G. Rehm, Forc@nslp2024: Overview and insights from the field of research classification shared task, in: G. Rehm, S. Dietze, S. Schimmler, F. Krüger (Eds.), Natural Scientific Language Processing and Research Knowledge Graphs, Springer Nature Switzerland, Cham, 2024, pp. 189–204.

[10] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: http://arxiv.org/abs/1908.10084.

[11] K. Wang, Y. Ding, S. C. Han, Graph neural networks for text classification: A survey, Artificial Intelligence Review 57 (2024) 190.
[12] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).
[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, arXiv preprint arXiv:1710.10903 (2017).
[14] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al., The llama 3 herd of models, arXiv preprint arXiv:2407.21783 (2024).
[15] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al., Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, arXiv preprint arXiv:2501.12948 (2025).
[16] G. Team, Gemma 3 Technical Report, Technical Report, Google DeepMind, 2025. URL: https://storage.googleapis.com/deepmind-media/gemma/Gemma3Report.pdf.

A. Prompts used in Gemma3

The following table contains the prompts used for the different scenarios as explained in the Methodology Section 4.3.

Table 3
Different prompts used with LLMs. The first prompt "Common" prepends each of the other variants in the list.

Variant	Prompt
Common	You are an NLP researcher writing an NLP-focused literature survey paper. You need to annotate papers with labels that are related to the topic in the manuscript. Only output a comma-separated list of labels for this manuscript. These are the details of the manuscript: Title = [title], Abstract = [abstract] ...
ZS-Flat	... Usually we have around 3 and at least 1 label. These are the available labels: [tax_string_flat]
ZS-Hier	... Usually we have Level 1: 1 to 9 labels, the median being 3, Level 2: 0 to 7 labels, the median being 2, Level 3: 0 to 3 labels, the median being 0. This is the hierarchical label taxonomy: [tax_string_hier].
NL	... Select among these labels from similar, labeled papers to the manuscript [neighbor_labels].
NL-Count	... Select among these labels from similar, labeled papers to the manuscript, with their relevance score [neighbor_labels_counts]