Enhancing Extreme Multi-Label Text Classification: Addressing Challenges in Model, Data, and Evaluation

Dan Li[1], **Zi Long Zhu**[1], **Janneke van de Loo**[1], **Agnés Masip Gómez**[2], **Vikrant Yadav**[1], **Georgios Tsatsaronis**[1], **Zubair Afzal**[1]
Elsevier[1], Univerisity of Amsterdam[2]
{d.li1, z.zhu, g.tsatsaronis, zubair.afzal}@elsevier.com
{Vikrant4.k, jannekevandeloo, agnesmgomez}@gmail.com

Abstract

Extreme multi-label text classification is a prevalent task in industry, but it frequently encounters challenges in terms of machine learning perspectives, including model limitations, data scarcity, and time-consuming evaluation. This paper aims to mitigate these issues by introducing novel approaches. Firstly, we propose a label ranking model as an alternative to the conventional SciBERT-based classification model, enabling efficient handling of large-scale labels and accommodating new labels. Secondly, we present an active learning-based pipeline that addresses the data scarcity of new labels during the update of a classification system. Finally, we introduce ChatGPT to assist with model evaluation. Our experiments demonstrate the effectiveness of these techniques in enhancing the extreme multi-label text classification task.

1 Introduction

Extreme Multi-label Text Classification (XMTC) refers to the task of assigning to each document its most relevant labels from a taxonomy, where the number of labels could reach hundreds of thousands or millions (Liu et al., 2017). XMTC plays a crucial role in various industry applications such as search systems, recommendation systems, and social media analysis. By enabling accurate categorization of documents, it facilitates making it easier to search, filter, and organize the content effectively (Li et al., 2022).

However, the existing approaches often face inherent challenges pertaining to the model, data, and evaluation aspects. First, classification models typically serve as the default choice for this task (Liu et al., 2017; Minaee et al., 2021). Nonetheless, these models struggle to scale to a large number of labels as the increasing size of feature space causes the number of parameters to explode quickly. Second, when building a new classification model, labeled data is often unavailable, and the available data can be imbalanced. Moreover, our taxonomy data, from which the labels originate, undergoes yearly updates. Consequently, both the training and test data, as well as the model, require regular updates. Third, the evaluation process is time-consuming. Evaluations are typically performed offline using a test set, which necessitates Subject Matter Expertss (SMEs) to spend significant time labeling samples. These existing issues have direct consequences for businesses, leading to prolonged release times, limited innovation, increased efforts for the sales team, and dissatisfied clients.

In this work, we aim to replace our existing classification pipeline with a new solution that addresses the aforementioned issues. First, we introduce a label ranking model to replace the SciBERT-based classification model used in production. This new model comprises a Bi-Encoder model and a Cross-Encoder model (Karpukhin et al., 2020; Craswell et al., 2021). The Bi-Encoder model offers benefits such as high recall and low computational cost, while the Cross-Encoder model enhances precision by re-ranking the top (i.e., 100) documents. Second, we propose an active learning-based pipeline for model updates and data collection. Since active learning needs an initial pool of positive documents, we use an unsupervised training strategy to train a Bi-Encoder that can adapt to our target domain. For new labels without labeled data, we use this Bi-Encoder model to identify potentially positive documents for annotation. Human annotators are then involved in the annotation loop to label the training data. Finally, we introduce ChatGPT to assist with model evaluation. We generate prompts for documents that require annotation and utilize ChatGPT (OpenAI) to obtain label answers along with confidence scores and explanations. Subsequently, SMEs manually verify these answers.

We assess our pipeline's performance by considering model effectiveness, training costs, and

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 313–321
December 6-10, 2023 ©2023 Association for Computational Linguistics

manual annotation costs. The predicted labels of our pipeline exhibit greater correctness and specificity compared to the production baseline. For a newly introduced label, it requires on average 100 human-annotated samples for the updated model to achieve a Recall@10 of 0.8. Additionally, with the help ChatGPT, SMEs' annotation effort is reduced from 15 mins to 5 mins for annotating a single document with 10 labels. As a result, our proposed pipeline enables multiple releases within a single year, significantly enhancing efficiency and productivity.

2 Related work

In the field of multi-label text classification, numerous studies have contributed to the development of effective models and techniques (Jiang et al., 2021; Yu et al., 2022). Previous research has explored a variety of methodologies, including traditional machine learning algorithms, deep learning architectures, and hybrid models, to address the complex nature of multi-label classification tasks (Chen et al., 2022). Notable work has been conducted on feature engineering (Scott and Matwin, 1999; Yao et al., 2018), neural network architectures (Onan, 2022; Soni et al., 2022), and loss functions tailored for multi-label scenarios (Hullermeier et al., 2020), aiming to enhance the predictive accuracy and interpretability of models. Furthermore, recent advancements in pre-trained language models, such as BERT (Devlin et al., 2019) and its variants (Zhuang et al., 2021), have demonstrated substantial results in multi-label classification, opening up new possibilities for transfer learning in this domain. Additionally, research efforts have delved into handling imbalanced label distributions (Huang et al., 2021; Xiao et al., 2021), leveraging auxiliary information, and adapting models for specific domains. The existing work provides a comprehensive foundation upon which our current research builds, with a focus on the capabilities of introducing new labels in a multi-label text classification setting.

3 Method

3.1 Label Ranking Model

We introduce a label ranking model to replace the SciBERT-based model in our cooperative production. It comprises a Bi-Encoder model and a Cross-Encoder model. The Bi-Encoder model offers benefits such as high recall and low computational cost, while the Cross-Encoder model enhances precision by re-ranking the top documents. See Figure 1 the architecture of our label ranking model.

A Bi-Encoder model (Karpukhin et al., 2020) employs a Siamese-Encoder architecture, where two sequences are encoded by the Transformer (Vaswani et al., 2017) in the same vector space separately, and their similarity is calculated upon their sequence embeddings. Similarly, our Bi-Encoder model consists of a document encoder and a label encoder, which are used to encode the document text and the label text separately. The two encoders share the same parameters. Each training batch contains only positive text pairs. To allow better negative sampling, we use the MultipleNegativesRanking loss (Oord et al., 2018; Henderson et al., 2017).

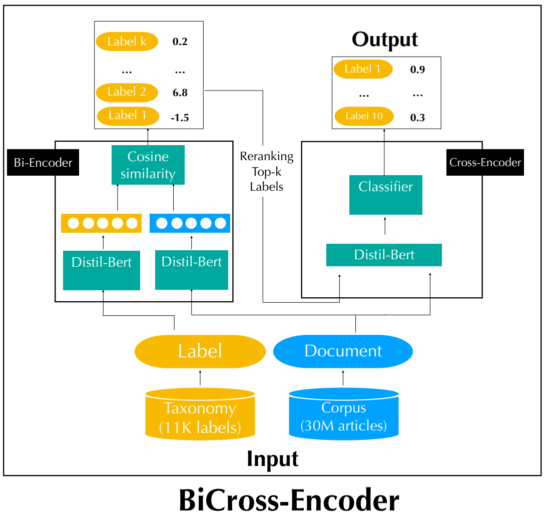


Figure 1: The architect of the Bi-Encoder and Cross-Encoder model.

Cross-Encoder (Craswell et al., 2021), a variant of the BERT classification model (Vaswani et al., 2017), has demonstrated state-of-the-art effectiveness in various IR tasks. However, it does not scale well for a large number of documents and is often applied after a Bi-Encoder. The Cross-Encoder takes as input the concatenated text "[CLS] label text [SEP] document text", which is processed by the encoder to model the semantic interaction among all pairs of tokens within the input sequence. Subsequently, the representation of "[CLS]" is then fed into a linear classifier and outputs a single score between 0 and 1 indicating how relevant the label is for the given document. For training examples, we create positive ones by using the ground truth labels of a document, and we create negative ones by randomly sampling 3 labels from the top 100 labels of the ranked list produced by Bi-Encoder. During inference we select the top 30 predictions from the Bi-Encoder for the Cross-Encoder to rerank to give our final prediction. The top 30 predictions from the Bi-Encoder are chosen using the nearest neigbour search algorithm using Hierarchical Navigable Small World (HNSW) graphs (Malkov and Yashunin, 2018) giving us a time complexity of $O(\log(|C|))$ during this selection process, with $|C|$ being the total number of labels[1].

An important detail during the training of the Bi-Encoder is to keep the same labels out of the same batch since the MultipleNegativesRankingLoss uses the other samples in the batch as negative examples. Therefore, if a label appears more than once it will create confusion due to samples from the same label acting as negative samples for each other.

3.2 Adapting BiCross-Encoder to New Labels

Industry taxonomies are dynamic, with new classes added and existing ones removed over time. Consequently, reclassifying existing and future documents using the updated taxonomy becomes necessary. The current standard practice involves fully retraining classification models from scratch after a taxonomy change, which is computationally inefficient and costly.

In this section, we illustrate a significant advantage of our label ranking model, as it allows for the introduction of new classes into the taxonomy without requiring full model retraining.

3.2.1 Cold-start Pool-based Active Learning

In the context of introducing a new label into a taxonomy, Active Learning (AL) provides an efficient approach to obtain labeled samples by iteratively learning from existing labeled samples and selecting unlabeled samples for annotation based on an acquisition strategy.

We perform the cold-start pool-based AL (Yuan et al., 2020) approach, which means we start with unlabeled samples denoted as U. We use an acquisition strategy $\mathcal{S}$ to select a subset U_s from U for annotation by an oracle $\mathcal{O}$ (SME annotators). Here we ask the oracle a binary question, i.e. given an unlabeled sample u, does it belong to class c. A model $\mathcal{M}$ iteratively learns a set of new labels C_{new} via the AL cycle as described in Algorithm 1.

Algorithm 1 Cold-start pool-based active learning cycle

```
Input: $\mathcal{O}, \mathcal{M}, \mathcal{S}$, U
1: for $i \leftarrow 1$ to $I$ do
2:     $U_s \leftarrow \mathcal{S}(\mathcal{M}, U)$
3:     $L \leftarrow \mathcal{O}(U_s)$
4:     $\mathcal{M} \leftarrow \text{train}(\mathcal{M}, L)$
5:     $U \leftarrow U \setminus U_s$
6: end for
```

3.2.2 Document Pool

In our corpus, a document can have multiple labels and therefore every document in the corpus is a potential candidate for newly introduced labels. The challenge here is that the corpus U_{corpus} has more than 14M documents and this requires practically infeasible computational resources to do model inference at each iteration of AL (line 3 in Algorithm 1). To address this challenge, we propose an alternative approach that utilizes a separate Bi-Encoder model to retrieve a relatively small number of potentially relevant documents, which serve as the unlabeled samples U, such that $|U| \ll |U_{\text{corpus}}|$.

To train the separate Bi-Encoder model, we select a random sample of 80K documents from the domain of the new labels, and then use the unsupervised domain adaptation method GPL (Wang et al., 2022) to finetune a pretrained Bi-Encoder model (*distilbert-base-uncased*). Using this GPL-trained model we select from U_{corpus} for each newly introduced label c, a subset U_c, by selecting the top 1000 documents that are semantically the closest to the label. Finally, U is defined as $U = \cup_{c \in C_{\text{new}}} U_c$.

3.2.3 Acquisition Strategy

The acquisition strategy is the key area of research within AL, however, these strategies are mostly based on classification-based models (Ren et al., 2021) and they are not directly suited for our label ranking model. For our task, we introduce a simple greedy acquisition strategy, where for each label $c \in C_{new}$ we rank the documents from U_c by their semantic similarity to the label of c according to the Bi-Encoder component of model $\mathcal{M}$. After the ranking, we uniformly sample a label $c \in C_{new}$ and take the top from the ranked U_c to be put in U_s. We perform this N times to create the batch U_s to be annotated by the oracle (SME annotators), as shown in Algorithm 2.

This strategy is greedy because we are forcing positive examples to be chosen for a given label

[1] We use the following python library: `https://github.com/nmslib/hnswlib`

c. In this scenario it is a valid heuristic, because the Bi-Encoder component in $\mathcal{M}$ learns using the MultipleNegativesRanking loss and this loss uses positive pairs as its input. So, it is necessary for our training process to find positive pairs between label and documents[2].

Algorithm 2 Greedy Acquisition Strategy $\mathcal{S}$

```
Input: $U, \mathcal{M}$
Output: $U_s$
$U_s \leftarrow \{\}$
for $c \in C_{\text{new}}$ do
    $U_c^{\text{ranked}} \leftarrow \text{rank}(\mathcal{M}, U_c)$
end for
for $i \leftarrow 1$ to $N$ do
    $c \leftarrow$ randomly sample from $C_{\text{new}}$
    $u \leftarrow top(U_c^{\text{ranked}})$
    $U_s \leftarrow U_s \cup \{u\}$
    $U_c^{\text{ranked}} \leftarrow U_c^{\text{ranked}} \setminus \{u\}$
end for
```

3.2.4 Model Training

Our first issue in training the model $\mathcal{M}$ is catastrophic forgetting, a phenomenon that occurs when learning new labels (Masana et al., 2020; Xia et al., 2021). This happens due to the given model adapting towards discriminating between the newly introduced labels without consideration for the decision boundaries towards the previously learned labels. An effective and straightforward solution is data replay (Masana et al., 2020), where data from the previous labels are included. We achieve this by random sampling batch instances U_{replay} and their labels from the whole corpus, where we have it with the samples annotated by the oracle $\mathcal{O}$, i.e. $U_s^{new} = U_s \cup U_{replay}$. We then use U_s^{new} as input to train the Bi-Encoder in model $\mathcal{M}$ with the MultipleNegativesRankingLoss in the AL cycle.

For the Cross-Encoder component of $\mathcal{M}$, we train it continuously together with the Bi-Encoder at each iteration. We first get the top k ranked documents from the updated Bi-Encoder, and then use the true label given by the oracle as a positive example and randomly sample 3 labels as the negatives, as mentioned in Section 3.1.

3.3 ChatGPT-assisted Evaluation

The absence of a test set presents a common challenge for offline evaluation. However, creating a test set can be a time-consuming task. For instance, providing SMEs with a single document and 10 labels can take approximately 15 minutes for annotation. The major reason is that SMEs are typically proficient in only one or two domains, and there is no expert who possesses knowledge across all domains. Even domain experts may lack comprehensive knowledge of highly specialized topics, making it difficult to precisely determine the relevance of a label to a given document. While ChatGPT has shown great potential to help data annotation in NLP (Gilardi et al., 2023; Thapa et al., 2023; Kuzman et al., 2023).

To address these challenges, we leverage ChatGPT as an assisting evaluation tool. We begin by generating prompts for the documents that require annotation and employ ChatGPT to provide label relevance scores (0=irrelevant, 1=somewhat relevant, or 2=highly relevant) along with explanations for these scores. Table 1 shows the prompt we used and the response from ChatGPT.

4 Web Interface

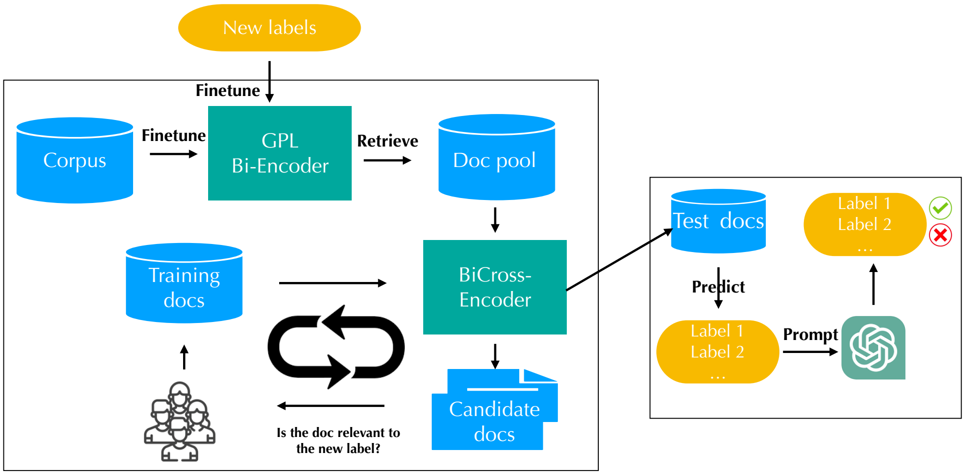
To facilitate efficient model updates and data annotation, we have developed a web application (Figure 3). This application enables multiple users to seamlessly interact with the model simultaneously, with all interactions logged and stored. It employs a microservices architecture for scalability, consisting of a front-end React client application and two FastAPI server applications. One server manages user and project management, while the other focuses on the AL component. Communication between the API and AL is facilitated through RabbitMQ message queues, and all data is stored in a MongoDB instance. The application can be hosted on a *p3.2xlarge* or a *g4dn.xlarge* Amazon EC2 instance.

At the beginning of the AL process, the BiCross-Encoder model provides a list of ranked documents by relevancy. These documents are shown one by one to all users without repetition. The users will be able to decide if the label matches the content of the abstract. Once a batch of positive results (label matches abstract) is obtained, it is sent to the model for training, and a new list of ranked abstracts is provided. The application's asynchronous nature ensures that users are unaffected by any time delays caused by these model processes. Additionally, user responses and time spent on annotations are stored and linked to project and abstract data.

[2] If a negative sample is found for a particular label, we simply skip this sample.

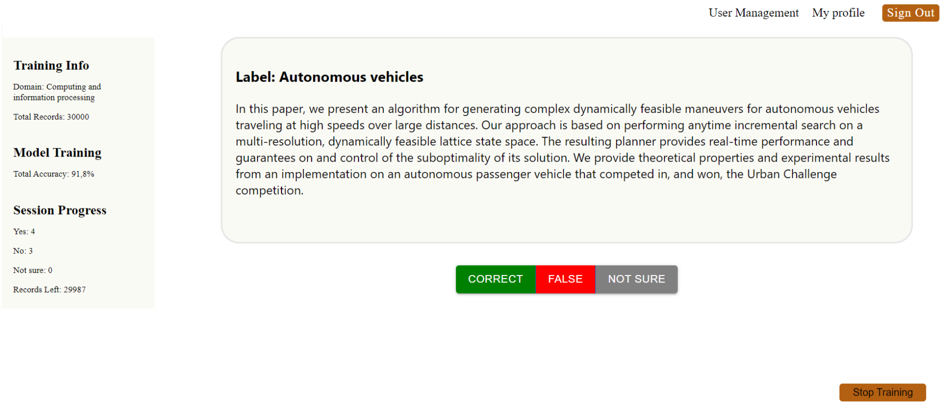
Prompt	Which of the following *0. Fuzzy neural networks ...* are relevant topics for this abstract. For each just provide a relevance score between 0 and 2, and an explanation. 0 means not relevant and 2 means highly relevant. -> *TITLE: ... ABSTRACT: ... the determination of the rail voltage for a 1500 V DC-fed rail system by means of the adaptive neuro-fuzzy inference system ...*
Response	*0. Fuzzy neural networks: 2 - The study uses an adaptive neuro-fuzzy inference system (ANFIS), which combines fuzzy logic and neural networks ...*

Table 1: An example for ChatGPT prompt and its response.



Model update & data collection via active learning **ChatGPT evaluation**

Figure 2: The architect of the pipeline: data collection, model update, and evaluation.



Training Info

Domain: Computing and information processing

Total Records: 30000

Model Training

Total Accuracy: 91,8%

Session Progress

Yes: 4

No: 3

Not sure: 0

Records Left: 29987

Label: Autonomous vehicles

In this paper, we present an algorithm for generating complex dynamically feasible maneuvers for autonomous vehicles traveling at high speeds over large distances. Our approach is based on performing anytime incremental search on a multi-resolution, dynamically feasible lattice state space. The resulting planner provides real-time performance and guarantees on and control of the suboptimality of its solution. We provide theoretical properties and experimental results from an implementation on an autonomous passenger vehicle that competed in, and won, the Urban Challenge competition.

Figure 3: The web interface of the pipeline.

The application also allows users to track model performance during their annotation, once they are satisfied with the performance they can terminate model training.

5 Experimental Setup

5.1 Data

Labels. The labels assigned to documents are derived from Elsevier's Compendex taxonomy, which encompasses approximately 11,486 labels from the generic engineering domain. This taxonomy exhibits a poly-hierarchy structure, wherein certain leaf nodes can have multiple parent nodes. The taxonomy undergoes regular updates, typically on an annual basis. These updates involve the addition of new labels and the potential removal of existing ones to ensure their accuracy over time.

Corpus. The corpus we work with contains about 14M documents of interdisciplinary engineering content. Each document has a title, an abstract, keywords, and some meta information; it is associated with several labels generated by a rule-based fuzzy string matching system. We use the concatenation of title, abstract, and keywords to encode the documents.

Document pool (DP) dataset. It consists of relevant and irrelevant documents for 7 taxonomy labels. For each label, the dataset contains between 250 and 450 documents (mean=363), which were manually annotated as relevant or irrelevant (mean=150 relevant documents). The irrelevant documents are mainly hard negatives.

Active learning (AL) dataset. Out of the 11,486 labels in the taxonomy, we randomly chose 30 labels to represent the newly introduced labels. Next, we utilized the GPL Bi-Encoder to select 1,000 samples for each concept from the corpus, resulting in an unlabeled pool of data comprising 30,000 samples. Additionally, we randomly selected a total of 5,000 documents from the dataset to form the test set for the 30 selected labels.

5.2 Baselines

Production model. The production model is a SciBert-based multi-label classification model, with a classification layer on top of the [CLS] output of the pre-trained SciBert model (*allenai/scibert_scivocab_uncased*). The classification model was finetuned using the MultiLabelSoftMarginLoss on a 2M documents subset of our 14M corpus, with taxonomy labels generated by a rule-based system.

6 Results

6.1 BiCross-Encoder Effectiveness

In this experiment, we aim to answer whether our label ranking model outperforms the classification model for extremely large label scenarios. The BiCross-Encoder model was trained on the 14M documents with weak labels generated by a rule-based system. The evaluation was done automatically using ChatGPT. We first select 22 documents from each of the 4 domains, i.e. communication, natural science, material science, and computer science; then we do inference using both models to produce a rank list from the 11,486 labels. We keep the top 10 labels and ask ChatGPT to answer whether the label is relevant to the corresponding document or not.

In Table 2, we find that BiCross-Encoder performs better than the SciBERT classifier in the domains of communication and computer science, and has comparable performance in natural science and material science.

A natural question about ChatGPT that readers might come up with is whether it is reliable for automatic evaluation. We manually ask SME to examine the answers (0, 1, or 2) from ChatGPT and give their own answer if the ChatGPT answer is not correct. The percentage of agreement is 60% on the original 3-point scale and 82% on a 2-point scale (mapping 1 and 2 as 1). The relatively low agreement from the 3-point scale is because of confusion between 1 (somewhat relevant) and 2 (highly relevant). Given that a 2-point scale is enough for most relevant tasks, we conclude that using ChatGPT for evaluation is acceptable if we are faced with limited time and monetary budget for annotation.

	# Correct labels / # All labels	
Domain	BiCross-Encoder	SciBERT
Communication	181/220	151/220
Natural Science	170/220	173/220
Material Science	164/220	185/220
Computer Science	180/220	105/220

Table 2: Performance of Bert classifier and BiCross-Encoder. The ground truth of the predicted labels was annotated automatically using ChatGPT.

6.2 GPL Bi-Encoder Effectiveness

In this experiment, we use the DP dataset to evaluate the ranking performance of the GPL-finetuned Bi-Encoder, which we use for selecting the initial document pool of potentially relevant documents. Figure 4 shows the effectiveness of ranking the relevant documents in the top-k, before and after finetuning with GPL. We are able to effectively finetune a pre-trained bi-encoder to the domain without any manual annotation effort. Since the goal is to retrieve as many potentially relevant documents as possible, we care about the recall score. We can see that with only 400 documents, the recall score reaches almost 100%.

To sum up, the finetuned model is well capable of selecting a set of relevant documents for a given label, consequently benefiting the efficiency of the AL loop.

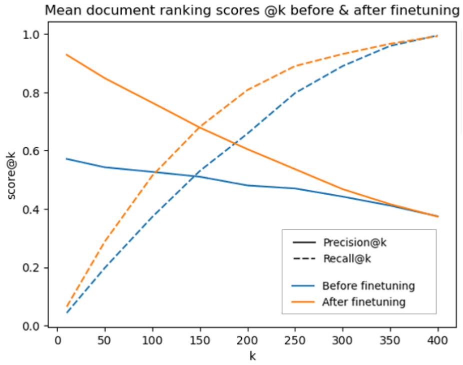


Figure 4: Effectiveness of the GPL finetuned Bi-Encoder in selecting potentially relevant documents for the initial pool.

6.3 Active Learning for New Labels

In this experiment, we show the results of AL the 30 newly introduced labels. Here we used a Bi-Encoder trained on the "old" labels and the *distilroberta-base* Cross-Encoder off the-shelf. The results are shown in Figure 5.

First, by training the Cross-Encoder to re-rank the Bi-Encoder label rankings, we observed a performance boost of approximately 15 points, resulting in a Recall@10 of 0.85. Second, the performance improvement was achieved with just 100 iterations. It is noteworthy that each iteration involved, on average, only 1 or 2 newly labeled samples, summing up to 100 samples per new label. This indicates that combining the selection of the initial pool via GPL and the greedy acquisition strategy together is a successful heuristic for newly introduced labels, especially in low-budget scenarios.

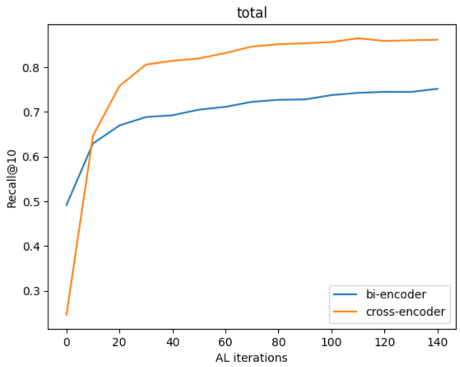


Figure 5: The performance in Recall@10 at each AL iteration.

	Before AL	After AL
Recall@10	0.4241	0.4852

Table 3: Performance of the Bi-Encoder before and after Active Learning on the "old" 11456 labels and on randomly sampled 50K documents from the corpus.

6.4 Active Learning Impact on Old Labels

Table 3 shows the performance of the Bi-Encoder before and after AL on the "old" taxonomy, i.e. excluding the newly introduced labels.

The result indicates that the model's performance remained consistent with the old labels even after applying AL. Surprisingly, the model's performance even exhibited a significant improvement. This finding confirms the efficacy of incorporating data replay as an effective countermeasure against catastrophic forgetting. Additionally, the integration of data replay in the Bi-Encoder model allowed it to learn the relation between the new and old labels in its semantic space. As a result, the embeddings between the old labels were better defined, leading to the observed enhanced performance following AL on the new classes.

7 Conclusion

In this work, we propose an approach to enhance our pipeline for the extreme multi-label text classification task. We replace the traditional SciBERT-based classification model with a label ranking model based on a Bi-Encoder and a Cross-Encoder, enabling efficient handling of large-scale labels. Moreover, we present an active learning-based pipeline that addresses the data scarcity of new labels during the update of a classification model. Finally, we demonstrate the effectiveness of using ChatGPT for model evaluation when faced with limited time and monetary budget for annotation.

Limitations

One of the limiting factors during the AL cycle is that our acquisition strategy is a greedy method. The acquisition strategies in existing works usually depend on the classification head and embedding space of a given model, which may not be directly compatible with our ranking-based model. A direction for future research would be looking at acquisition strategy for ranking-based models.

Another limitation is that in the AL cycle, only the positively annotated samples by the oracle are used for training the model. This is not entirely efficient because the negatively annotated samples are not used, while they also cost resources. A possible solution is to have a different loss that incorporates these negatively annotated samples during training. Another solution is to change the task of the oracle to give all the categories a sample belongs to.

References

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 *Artificial Intelligence and Security: 8th International Conference, ICAIS 2022, Qinghai, China, July 15–20, 2022, Proceedings, Part I*, page 443–456, Berlin, Heidelberg. Springer-Verlag.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. Ms marco: Benchmarking ranking models in the large-data regime. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1566–1576.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Yi Huang, Buse Giledereli, Abdullatif Köksal, Arzucan Özgür, and Elif Ozkirimli. 2021. Balancing methods for multi-label text classification with long-tailed class distribution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8153–8161, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Eyke Hullermeier, Marcel Wever, Eneldo Loza Mencía, Johannes Furnkranz, and Michael Rapp. 2020. A flexible class of dependence-aware multi-label loss functions. *Machine Learning*, 111:713–737.

Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. 2021. Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7987–7994.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Taja Kuzman, Igor Mozetic, and Nikola Ljubesic. 2023. Chatgpt: Beginning of an end of manual linguistic data annotation? use case of automatic genre identification. *ArXiv, abs/2303.03953*.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S Yu, and Lifang He. 2022. A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(2):1–41.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 115–124.

Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.

Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. 2020. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:5513–5533.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40.

Aytuğ Onan. 2022. Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification. *J. King Saud Univ. Comput. Inf. Sci.*, 34:2098–2117.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

OpenAI. Chatgpt.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40.

Sam Scott and Stan Matwin. 1999. Feature engineering for text classification. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 379–388, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Sanskar Soni, Satyendra Singh Chouhan, and Santosh Singh Rathore. 2022. Textconvonet: a convolutional neural network based architecture for text classification. *Applied Intelligence (Dordrecht, Netherlands)*, 53:14249 – 14268.

Surendrabikram Thapa, Usman Naseem, and Mehwish Nasim. 2023. From humans to machines: can chatgpt-like llms effectively replace human annotators in nlp tasks. In *Workshop Proceedings of the 17th International AAAI Conference on Web and Social Media*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2345–2360.

Congyin Xia, Wenpeng Yin, Yihao Feng, and P. L. H. Yu. 2021. Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system. In *North American Chapter of the Association for Computational Linguistics*.

Lin Xiao, Xiangliang Zhang, Liping Jing, Chi Huang, and Mingyang Song. 2021. Does head label help for long-tailed multi-label text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14103–14111.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2018. Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. *BMC Medical Informatics and Decision Making*, 19.

Hsiang-Fu Yu, Kai Zhong, Jiong Zhang, Wei-Cheng Chang, and Inderjit S Dhillon. 2022. Pecos: Prediction for enormous and correlated output spaces. *Journal of Machine Learning Research*.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.