Scalable Patent Classification with Aggregated Multi-View Ranking

Dan Li*, **Vikrant Yadav***, **Zi Long Zhu***,
Maziar Moradi Fard, Georgios Tsatsaronis, Zubair Afzal
Elsevier,
Radarweg 29, Amsterdam, 1043NX, The Netherlands
vikrant4.k@gmail.com, {d.li1, z.zhu, m.moradifard, g.tsatsaronis, zubair.afzal}@elsevier.com

Abstract

Automated patent classification typically involves assigning labels to a patent from a taxonomy, using multi-class multi-label classification models. However, classification-based models face challenges in scaling to large numbers of labels, struggle with generalizing to new labels, and fail to effectively utilize the rich information and multiple views of patents and labels. In this work, we propose a multi-view ranking-based method to address these limitations. Our method consists of four ranking-based models that incorporate different views of patents and a meta-model that aggregates and re-ranks the candidate labels given by the four ranking models. We compared our approach against the state-of-the-art baselines on two publicly available patent classification datasets, USPTO-2M and CLEF-IP-2011. We demonstrate that our approach can alleviate the aforementioned limitations and achieve a new state-of-the-art performance.

Keywords: multi-label classification, dense retrieval, reranking, graph model, L2R

1. Introduction

The life cycle of a *patent* involves several steps and roles: a patent attorney drafts a patent application; a patent officer classifies the application with a hierarchical *patent classification schema* such as International Patent Classification (IPC) or Cooperative Patent Classification (CPC); a patent examiner assesses the patentability of the described invention; and finally, the decision of being granted or not is made (Krestel et al., 2021). An important part of this procedure is the classification of patents, which helps to allocate appropriate experts to review the patent application and minimize the search effort. Automated patent classification helps to reduce the classification effort. It is often formulated as a multi-label classification task in the current literature (Roudsari et al., 2021; Fang et al., 2021; Lee and Hsiang, 2020).

Previous works in the realm of patent classification have predominantly centered on refining the *subclass level* of the IPC/CPC taxonomy, which encompasses approximately 600 labels. Most works have a classification layer on top of their models. However, classification-based models suffer from several drawbacks and hinder the wide application of these models. Firstly, classification-based models can not generalize to newly introduced labels, due to the fixed dimension of the output probability distributions. This is an important issue because patent classification schemas are often changed, by adding new labels and by removing and redefining existing labels. Secondly, the classification layer of these models does not scale to a large number of labels. For example, the IPC schema has around 600 labels at the subclass level and around 300,000 at the subgroup level. To accommodate this many labels, the model size can become unsustainable due to the required feature and output space. In most of the current literature, the main focus is therefore on the subclass level. Thirdly, a classification-based model cannot utilize the textual descriptions of the labels themselves.

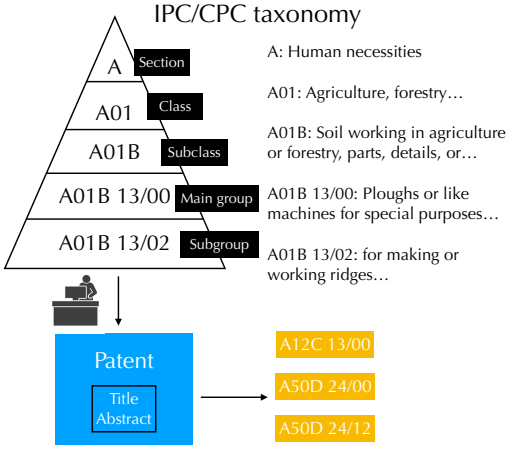

Figure 1: This figure illustrates the task of multi-label patent classification, wherein a domain expert is responsible for assigning multiple CPC/IPC labels to a patent.

In this work, we alleviate these drawbacks by formulating the classification of patents as a ranking problem. For this purpose, we introduce a multi-view model to classify Patents with Aggregated Ranking of Labels (we name it PARL). It consists of four different component models with an aggregation model on the top. Each of the component models aims to retrieve a small number of label candidates from the whole label set. The first two

* These authors contributed equally to this work.

LREC-COLING 2024, pages 14336–14346

component models are the Bi-Encoder and Cross-Encoder, which use the *patent-label view*. The third and fourth component models BM25 and GraphSAGE (Hamilton et al., 2017) use the *patent-patent view*. The model on top is a simple Learning-to-Rank (L2R) model, which aggregates the retrieved labels from the component models and ranks the subset of labels based on the component ranking scores.[1]

The combination of the individual components results in a novel and efficient approach tailored to the patent classification task. Our method has the following merits. First, **consistent model size for scaling labels**. The four component models including BE, CE, GS, and BM25 remain constant in size regardless of the dataset's label number. This unique feature underscores the scalability of our approach without compromising on model size. Second, **efficient adaptation to taxonomy changes**. The dynamic nature of patent taxonomies is a challenge in the field. Unlike some previous methods that necessitate complete model retraining, we can simply train on newly introduced labels, reducing the time and computational resources required. Third, **production-friendly approach**. The BE retrieves the top 30 labels and the CE reranks these filtered labels. This design enhances efficiency by narrowing down the ranking scope and focusing computational efforts on the most relevant labels. Additionally, GS learns patent embeddings with a small-size neural network and retrieves similar patents by searching local neighborhoods in the graph. Fourth, **flexible integration of customized component models.** The L2R model takes the predicted scores of multiple components as features, thus it is easy to integrate new component models by adding extra features to the L2R model.

To sum up, our work introduces a novel approach to patent classification that addresses scalability, efficiency, and adaptability to evolving taxonomies. The main contributions are as follows:

- **Proposal of a new multi-view label ranking model**. Our model captures the diverse perspectives present in patents by using different component models. It is one of the first attempts to address this taxonomy at a lower level, involving a significantly larger number of labels. Moreover, its flexibility enables easy integration of diverse component models in a customized scenario.
- **Superior performance over state-of-the-art models**. The model outperforms prior state-of-the-art models on two public datasets USPTO-2M and CLEF-IP-2011.
- **Comprehenstive analysis of model effectiveness**. We conduct a comprehensive analysis to unravel the underlying mechanisms driving the effectiveness of our model. This analysis encompasses the impact of different views, visualization of the embeddings, and the difference in label prediction from different views.

[1] Please note that we interchangeably use the abbreviations BE, CE, and GS to refer to Bi-Encoder, Cross-Encoder, and GraghSAGE throughout this paper.

2. Related Work

2.1. Patent Classification

Recent works on patent classification have focused on using deep learning techniques to improve the efficiency and accuracy of the classification process. Representation models such as convolutional neural networks (Li et al., 2018a), recurrent neural networks (Xiao et al., 2018), and transformers (Li et al., 2022; Lee and Hsiang, 2020) represent patent texts into embeddings and classify patents. They have shown improved performance compared to traditional machine learning methods, such as support vector machines and Naive Bayes (Chu et al., 2008). Incorporating external knowledge sources, such as WordNet and Wikipedia, has been shown to be effective in providing valuable contextual information that can enrich the understanding of patent documents (Al-Shboul and Myaeng, 2011). Additionally, attention mechanisms have been employed to identify the most relevant parts of the patent document for classification (Haghighian Roudsari et al., 2020). Moreover, graph convolutional networks and graph attention networks have been utilized to model the relationships between different patents, leading to improved classification performance (Tang et al., 2020). Leveraging the inherent structural information present in patent data, these graph-based approaches enable a more comprehensive analysis of the interconnections and dependencies among patents.

Overall, these recent works have demonstrated their efficiency and accuracy in patent classification. However, more research is needed to investigate the scalability and robustness of these methods in real-world scenarios.

2.2. Multi-view Learning

Multi-view learning is a machine learning approach that focuses on problems where each data instance benefits from multiple perspectives or sources of information. A concrete example of multi-view is a video that contains audio and visual information. Similarly, for a piece of text, one view could be lexical representations using a bag-of-words approach, while another view could be semantic embeddings. Multi-view learning has been used as an effective technique in the cases of texts (Fang et al., 2021), images (Seeland and

Mäder, 2021), and videos (Cai et al., 2019; Cui et al., 2018).

A notable application of multi-view learning in patent classification is Patent2Vec (Fang et al., 2021). This work leverages multi-view patent graph analysis to enhance the classification process. By employing techniques such as graph representation learning, view enhancement, attention-based multi-view fusion, and view alignment, Patent2Vec aims to improve classification accuracy and enhance interoperability using both metadata and textual information. In our work, we exploit different signals for each patent document, encompassing lexical and semantic contents within individual patents, as well as inter-patent signals linking different patents. The multi-view representation is a means to address the challenge of data imbalance. Specifically, a Bi-Encoder model and a Cross-Encoder model can learn good semantic representation for labels with sufficient samples, while the GraphSAGE model can provide effective representation for labels with limited samples.

3. Method

3.1. Rationale behind PARL

We observe that the data (USPTO-2M & CLEF-IP) we worked on contains patents from various domains, each with its specific terminologies and semantic patterns. The multi-view approach can uncover latent patterns across different views and capture the diverse relationships between the patents and their labels.

In the realm of capturing semantic signals between queries and documents, both Bi-Encoder and Cross-Encoder models have achieved remarkable success(Wang et al., 2022). However, an alternative approach utilizing graph-based networks, such as Text Graph Convolutional Network (Yao et al., 2018), has shown promise by exploiting word co-occurrence and document-word relationships to construct informative embeddings for words and documents. This technique has demonstrated effectiveness in showing the relationship between word-word, word-document, and document-document in the latent space. Additionally, empirical evidence has indicated that BM25 as a lexical retrieval method can surpass Bi-Encoder models in the cases of zero-shot scenarios and instances where an exact match is essential, such as keyword or entity retrieval (Zhao et al., 2022). These distinct types of signals can be leveraged effectively to capture patent-label and patent-patent relationships.

Figure 2 shows the multi-view approach we proposed. It consists of a Bi-Encoder and Cross-Encoder for semantic matching of patents and labels, BM25 and GraphSAGE for label retrieval through patent matching, and a L2R model that employs input vectors comprising scores of labels retrieved from the four aforementioned models, and handcrafted features for the label.

3.2. Bi-Encoder

Following (Karpukhin et al., 2020), the Bi-Encoder model consists of a patent encoder and a label encoder, which are used to encode the patent text and the label text separately. The two encoders share the same parameters. We use *msmarco-distilbert-dot-v5*[2] as the starting encoder model to be finetuned. Each training batch contains only positive text pairs, i.e. $B = \{(p_i, l_i) \mid i = 0, \ldots, |B|\}$, where l_i is a correct label of patent p_i. To allow better negative sampling, we use the MultipleNegativesRanking loss (Oord et al., 2018; Henderson et al., 2017).

$$\mathcal{L}^{BE} = -\frac{1}{|B|}\sum_{i=1}^{|B|}\log\frac{e^{\langle \boldsymbol{v}_{p_i}, \boldsymbol{v}_{l_i}\rangle}}{\sum_{j=1}^{|B|} e^{\langle \boldsymbol{v}_{p_i}, \boldsymbol{v}_{l_j}\rangle}} \quad (1)$$

where $\langle\rangle$ is the dot product, $\boldsymbol{v}_{p_i}$ and $\boldsymbol{v}_{l_i}$ are the embeddings of patent p_i and label l_i.

The Bi-Encoder model is beneficial for high recall and low computational cost. For training, the attention of the Transformer is within short texts of the patent or label and thus fast; for inference, the label embeddings are computed offline and only the embedding of the target patent needs to be computed online.

3.3. Cross-Encoder

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 input of the Cross-Encoder is the concatenated text "[CLS] label text [SEP] patent text". It is fed into the encoder for modeling the semantic interaction between any two tokens of the input sequence. The representation of "[CLS]" is then input to a linear classifier to output a single score between 0 and 1 indicating how relevant the label is for the given patent. We use *ms-marco-MiniLM-L-6-v2*[3] as the starting model to be finetuned. For training examples, we create positive ones by using the ground truth labels of a patent, and we create negative ones by randomly sampling 3 labels from the ranked list of Bi-Encoder's top 30 labels.

[2]https://huggingface.co/sentence-transformers/msmarco-distilbert-dot-v5

[3]https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

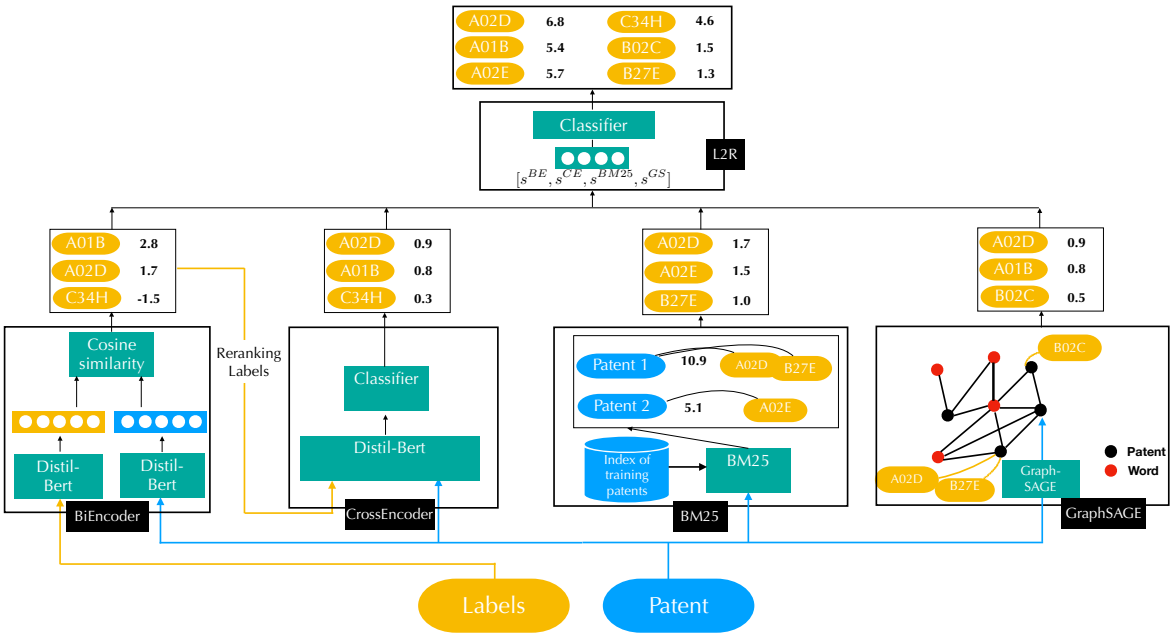

Figure 2: The architecture of the PARL model. It comprises Bi-Encoder and Cross-Encoder to capture the patent-label view, BM25 and GraphSAGE to capture the patent-patent view, and an L2R model to aggregate the retrieved labels from the component models and rank these labels based on the component ranking scores.

3.4. BM25

BM25 (Robertson et al., 2009) is a widely used model for lexical retrieval. We use it to discover lexically similar patents of a given patent.

For a given patent p, we take it as a query and use the BM25 algorithm to retrieve patents in the training data. For a retrieved patent, denote a tuple (p_i, s_i, L_i) as the patent, the BM25 score, and the set of ground truth labels. Thus, the relevant score of any label l to a given patent p is defined as

$$s(l, p) = \sum_{i=1}^{k} s_i \cdot \mathbb{I}(l \in L_i) \quad (2)$$

For BM25 it is not guaranteed to get k unique labels, since the top matching patents might have duplicate labels. Therefore, we take the top 50 matching patents to ensure that enough number of labels are present.

3.5. GraphSAGE

GraphSAGE (Hamilton et al., 2017) is an inductive model, which generates node embeddings by leveraging the structural information between different nodes, represented as a graph. The node embeddings are generated by learning a function that aggregates feature embeddings provided by the neighborhood for a given node.

Here we create a graph G representing an indirect patent-to-patent relationship. The nodes in the graph are patents and words. We only keep content words (nouns, verbs, and adjective words) for each patent. The structure of G is identical to the semantic graph used in Fang et al. (2021), where two words are connected if they have a positive Point-wise Mutual Information (PMI) score. Patent and word nodes have an unweighted connection if the word is present in the patent.

First, we generate word embeddings using FastText (Bojanowski et al., 2016) and represent the patent nodes as the average of the word embeddings. Then we train the GraphSAGE model on only the patent nodes with the same loss as Eq.(1). Lastly, we calculate the dot product between a given patent and all the patents to retrieve the top-k unique labels. The score of a label is calculated the same as Eq.(2).

3.6. Learning to Rank Labels

After each component model has ranked the top k labels for patent p, the set of chosen labels will be denoted as $L = \{l_1, ..., l_m\}$, where m denotes the total number of chosen labels. For each label $l_j \in L$ we create a corresponding feature vector,

$$x_j = [s_j^{BE}, s_j^{CE}, s_j^{BM25}, s_j^{GS}] \quad (3)$$

Here $s_j^{BE}, s_j^{CE}, s_j^{BM25}$ and s_j^{GS} are the scores given by Bi-Encoder, Cross-Encoder, BM25 and GraphSAGE respectively. If a component model did not rank a label in its top k, its score is set to 0.

Our L2R model is a generic linear network, which takes the feature x_j as input and outputs a score s_j. The training loss is the sped-up RankNet loss (Burges, 2010), calculated as

$$\mathcal{L}^{L2R} = \sum_{j,q \in |B|} [\frac{1}{2}(1 - s_{jq})\delta_{jq} - \log(1 + e^{-\delta_{jq}})] \quad (4)$$

where $B_{m \times 4}$ is the batch data; $\delta_{jq} = s_j - s_q$ and $s_{jq} \in \{1, 0, -1\}$, such that $s_{jq} = 1$ or 0 or -1 means label l_j should be ranked higher or equally or lower than l_q according to the true rank order. Using the s_j scores we create our aggregated ranking of labels.

4. Experimental Setup

4.1. Research Questions

We aim to examine the potential of our label ranking model in addressing the inherent limitations of classification-based models, as well as exploring the multi-views on label prediction. We formulate the following research questions. (RQ1) To what extent does the proposed model outperform the baseline models? (RQ2) How effective is the proposed model when faced with a large number of labels? (RQ3) How effective is the proposed model when faced with zero-shot labels? (RQ4) What is the impact of different views on model performance?

4.2. Datasets

USPTO-2M. The dataset (Li et al., 2018b) contains 2 million granted patents from the online website of the United States Patent and Trademark Office (USPTO). Each patent comprises a title, an abstract, and a set of class labels from the subclass level of IPC. As the Bi-Encoder component model needs label text as the input, we extract the definitions of the subclass levels from *patentview*[4], where each label has a keywords-like definition describing the most important technical part.

USPTO-1K/5K/10K. To evaluate our model on large-scale labels, we create three new datasets by extending USPTO-2M with subgroup labels. The subgroup labels are from *patentview*, which is an open-sourced and pre-processed version of the official USPTO patent bulk data. We restrict the labels under the G class (Physics) and select one thousand, five thousand, and ten thousand most frequent labels for the three datasets, respectively.

The selection of the G class was made for the following consideration. There are 9 sections in the CPC taxonomy, including 'G', each mirrors the hierarchical structure in the complete taxonomy (section, class, subclass, main group, and subgroup). By sampling from a single section, we ensured that our dataset encompasses labels from all five hierarchical levels, thereby maintaining a distribution analogous to the full CPC taxonomy. Furthermore, a single section presents a more challenging task due to the similarity of labels within this specific domain, thereby testing our model's efficacy in a more difficult classification scenario.

USPTO-5K-zeroshot. To mimic the zero-shot setting, we use USPTO-1K as the training data to train the model and evaluate it trained on a subset of USPTO-5K. This new test set has removed the overlapped labels with USPTO-1K, so that the labels in the test set do not appear in the training set.

CLEF-IP-2011. The dataset (Piroi et al., 2011) contains a mix of patents in English, French, and German. Not all patents include citation data. Therefore, we only select patents that are in English and include the necessary data fields. This resulted in a total of 187,812 patents.

4.3. Baselines

FastText We selected FastText as a baseline for its ability to serve as a label ranking benchmark, the same as our method's focus. FastText is a dense retrieval model with a different encoding approach than our BiEncoder component, providing a useful comparison. Empirically, FastText not only demonstrates robust performance but, in some scenarios, surpasses PatentBert in smaller datasets (USPTO-1K, USPTO-10K), establishing it as a strong baseline. The inclusion of FastText is not arbitrary but provides a strong baseline and ensures a comprehensive evaluation against varied approaches. We use the *FastText* library[5] to produce the embeddings for labels and patents, then for each patent, we rank the labels based on the cosine similarity between the label embedding and the patent embedding.

PatentBert (Lee and Hsiang, 2020). PatentBert is a Bert-based multi-label classification model. It has shown better performance than many patent classification models such as DeepPatent (Li et al., 2018b) and thus we compare our model with it. It fine-tunes the pre-trained *bert-base-uncased* model using the binary cross entropy loss. We implemented the model ourselves. The batch size is 32 and the maximum sequence length is 512. We train the model for 5 epochs.

Patent2Vec (Fang et al., 2021). Patent2Vec is a multi-view multi-label classification model. Patent2Vec uses three different views. The first one is the semantic view, which is identical to our GraphSAGE component model. The second and third views create a graph using patent citations

[4] https://patentsview.org/download/data-download-tables

[5] https://github.com/facebookresearch/fastText

Name	Training		Test		Level	Taxonomy
	# Example	# Label	# Example	# Label		
USPTO-2M	1,600,117	631	200,015	618	subclass	IPC
CLEF-IP-2011	150,249	609	28,172	589	subclass	IPC
USPTO-1K	116,935	971	14,520	971	subgroup	CPC
USPTO-5K	331,470	4,990	40,888	4,895	subgroup	CPC
USPTO-10K	466,340	9,988	57,670	9,696	subgroup	CPC
USPTO-5K-zeroshot	-	-	11,803	3,248	subgroup	CPC

Table 1: Statistics of the datasets used throughout the paper. USPTO-2M and CLEF-IP-2011 are widely used datasets on the subclass labels of IPC. USPTO-1K/5K/10K and USPTO-5K-zeroshot are datasets created by us for experiments on large-scale and zero-shot labels.

Method	R@1	R@3	R@5	R@10	P@1	P@3	P@5	P@10	NDCG@1	NDCG@3	NDCG@5	NDCG@10
USPTO-2M dataset												
FastText	0.49	0.74	0.82	0.89	0.66	0.33	0.22	0.12	0.49	0.58	0.59	0.61
PatentBert	0.58	0.81	0.87	0.93	0.68	0.34	0.22	0.12	0.68	0.75	0.78	0.80
Patent2Vec	0.56	0.79	0.87	0.93	0.65	0.33	0.22	0.12	0.65	0.73	0.77	0.79
LightXML	0.55	0.75	0.82	0.89	0.64	0.31	0.21	0.12	0.64	0.70	0.73	0.76
PARL (ours)	**0.61**	**0.84**	**0.91**	**0.95**	**0.71**	**0.36**	**0.24**	**0.13**	**0.72**	**0.78**	**0.81**	**0.83**
CLEF-IP-2011 dataset												
FastText	0.45	0.69	0.77	0.85	0.71	0.41	0.29	0.16	0.70	0.69	0.72	0.75
PatentBert	**0.56**	0.81	**0.89**	0.93	**0.86**	**0.48**	**0.33**	**0.18**	**0.86**	**0.83**	**0.85**	**0.87**
Patent2Vec	0.50	0.75	0.83	0.90	0.76	0.45	0.31	0.18	0.76	0.75	0.78	0.81
LightXML	0.52	0.75	0.82	0.89	0.79	0.44	0.30	0.17	0.79	0.76	0.78	0.81
PARL (ours)	0.55	**0.82**	**0.89**	**0.94**	0.84	**0.48**	**0.33**	**0.18**	0.84	**0.83**	**0.85**	**0.87**
USPTO-1K dataset												
FastText	0.31	0.53	0.63	0.75	0.46	0.29	0.22	0.13	0.46	0.50	0.54	0.59
PatentBert	0.29	0.51	0.61	0.73	0.43	0.28	0.21	0.13	0.43	0.48	0.52	0.57
Patent2Vec	0.30	0.52	0.61	0.73	0.48	0.32	0.23	0.15	0.48	0.51	0.54	0.59
LightXML	0.19	0.36	0.47	0.62	0.3	0.2	0.16	0.11	0.30	0.34	0.38	0.43
PARL (ours)	**0.39**	**0.63**	**0.71**	**0.79**	**0.58**	**0.35**	**0.25**	**0.14**	**0.58**	**0.61**	**0.64**	**0.67**
USPTO-5K dataset												
FastText	0.22	0.40	0.49	0.61	0.42	0.29	0.22	0.15	0.12	0.16	0.17	0.18
PatentBert	0.24	0.43	0.52	0.65	0.45	0.31	0.24	0.16	0.45	0.45	0.48	0.53
Patent2Vec	0.20	0.37	0.46	0.59	0.42	0.29	0.23	0.16	0.42	0.41	0.44	0.48
LightXML	0.20	0.36	0.45	0.57	0.37	0.25	0.2	0.13	0.37	0.38	0.40	0.45
PARL (ours)	**0.30**	**0.53**	**0.61**	**0.69**	**0.56**	**0.39**	**0.29**	**0.17**	**0.56**	**0.57**	**0.55**	**0.59**
USPTO-10K dataset												
FastText	0.19	0.36	0.45	0.57	0.42	0.29	0.22	0.15	0.10	0.13	0.14	0.15
PatentBert	0.20	0.37	0.46	0.58	0.43	0.30	0.23	0.16	0.43	0.42	0.44	0.48
Patent2Vec	0.18	0.35	0.44	0.56	0.42	0.30	0.24	0.17	0.42	0.41	0.43	0.47
LightXML	0.18	0.32	0.40	0.51	0.36	0.25	0.19	0.13	0.36	0.36	0.37	0.41
PARL (ours)	**0.26**	**0.47**	**0.57**	**0.69**	**0.54**	**0.38**	**0.30**	**0.19**	**0.54**	**0.53**	**0.55**	**0.59**

Table 2: Performance comparison of our PARL model with baselines on USPTO-2M and CLEF-IP-2011 (IPC *subclass* labels), USPTO-1K/5K/10K (CPC *subgroup* labels) datasets. PARL achieves superior results on USPTO-2M, USPTO-1K/5K/10K and performs comparably to PatentBert on CLEF-IP-2011.

and assignees, or citations and inventors as link relations. These two views are both trained using metapath2vec (Dong et al., 2017). Finally, a multi-label classification model is trained such that, each view is first enhanced by adding features from the different views combined, and then the three views are combined using an attention mechanism.

LightXML (Jiang et al., 2021). Since the proposed approach was also tested on a large number of labels we wanted to evaluate its effectiveness when compared to extreme classification models. We use LightXML as a representative baseline for extreme multi-label text classification models. It uses a transformer model to represent patents and TF-IDF vectors to represent labels and proposes a generative cooperative network that first retrieves a small set of labels and then reranks those labels. We tuned the hyperparameters on the validation set to search for the best performance of the model.

4.4. Evaluation Metrics

Since we model the multi-label patent classification problem as a label ranking task, we use ranking-oriented metrics. We report P/R/NDCG@k, k=1,3,5,10. Note that P@1 is more informative than P@3/5/7 because, in our datasets, patents have on average 1 to 2 labels.

4.5. Implementation Detail

We use the title and abstract as the text of a patent and set the text sequence length to 512 WordPiece tokens for the Bi-Encoder and Cross-Encoder. For each component model, we take the top $k = 10$ predicted labels for the aggregated ranking performed by L2R. The training batch size is 32, 16, and 128 for Bi-Encoder, Cross-Encoder, and GraphSAGE.

Note that the training of every model is performed on a single Nvidia Tesla V100 GPU with 16 GB of memory and 61 GB of memory on the CPU (p3.2 instance on AWS). We intentionally make the models light so that they can be applied across a wide range of tasks. The code for our model and experiments is publicly available[6].

5. Results

5.1. Main Results

This experiment aims to answer RQ 1. Following the same setting as previous works, we evaluate our model on two public datasets, USPTO-2M and CLEF-IP-2011. These two datasets have subclass-level labels. The results are presented in Table 2. Our PARL model achieves the best performance among the baselines by a large margin on USPTO-2M. On CLEF-IP-2011, our model performs similarly to PatentBert and better than the other models.

An important observation is that our baselines and our own model all are capable of running on an AWS p3.2xlarge instance (a single GPU with 16 GB memory and 61 GB of CPU memory) and we did not consider any baseline that could not run on this system. For example, XR Transformer(Zhang et al., 2021) is a state-of-the-art extreme multi-classification model, however, it requires a p3.16xlarge AWS instance (64 CPUs with 488 GB memory and 8 GPUs with 128 memory) for training. XR Transformer can beat most of our baselines when using a larger system. However, if we compare XR Transformer with LightXML in Zhang et al. (2021), then XR Transformer only outperforms LightXML by a few points on various datasets, whereas our PARL model outperforms LightXML more significantly on the patent datasets.

[6] https://github.com/dli1/parl

5.2. Large-scale Labels

To our best knowledge, the existing literature on patent classification has been mainly focused on the subclass labels from IPC/CPC. With only a few works reporting on main group labels (Zuo et al., 2022) and none on the subgroup labels. This experiment aims to see if our proposed model is able to scale towards the subgroup labels (RQ2).

Results are presented in Table 2. PARL performs the best on the USPTO-1K/5K/10K dataset. The performance for all models decreases worse as the number of labels increases and the baseline model decreases faster. The loss of performance for PatentBert is the worst, as its performance becomes worse than that of the component models from PARL. This indicates that the number of features becomes too small to discriminate between large labels. To accommodate, it is possible to increase the number of parameters. However, this is infeasible as there can be 100,000 or more labels. Patent2Vec does not seem to suffer as much from this issue. This is likely due to the unsupervised trained embeddings. Creating a well-defined feature space, such that the burden of discriminating between labels becomes less for the classification head.

5.3. Zero-shot Labels

We conducted this experiment to understand how well the component model Bi&Cross-Encoder adapts to new labels (RQ 3). We did not report the baselines or BM25 and GraphSAGE as they are not applicable in this scenario.

Table 3 shows that Bi&Cross-Encoder is able to achieve an R@10 of 0.41 and NDCG@10 of 0.26. It is quite impressive given that the label set consists of more than 3000 labels the model has never seen. It is a common practice to collect data for new labels, and the model can be used to collect potential positive patents to save annotation costs, or to serve as the initial model for an active learning loop for data collection.

5.4. Multi-view Impact

In this experiment, we conducted an ablation study of PARL to gain insights into the contributions from different views. This is achieved by excluding one and two component models and evaluating the performance impact on the USPTO-2M dataset. The results are illustrated in Figure 3.

Our first important observation is that removing any individual component model from PARL results in a performance loss. This shows that the different component models provide different views that positively impact the performance of the L2R model. However, not each model contributes equally. For

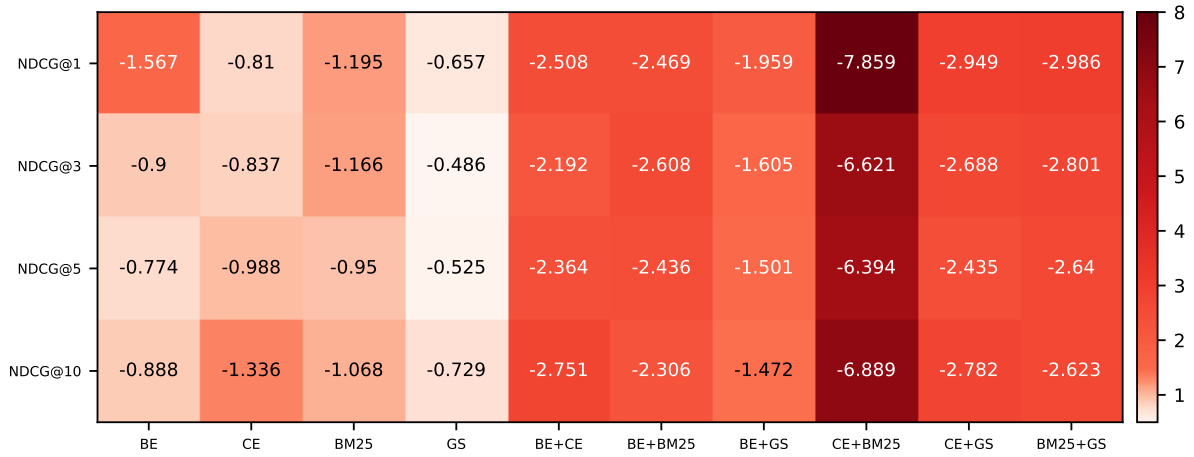

Figure 3: The significance of each model in PARL, by evaluating the performance impact of excluding one or two component models NDCG ($\times$100) on the USPTO-2M dataset. Notably, the omission of the CE and BM25 models leads to the most substantial performance drop, indicating their crucial contribution towards PARL.

Method	R@10	NDCG@3	NDCG@5	NDCG@10
Bi&Cross-Encoder	0.41	0.19	0.22	0.26

Table 3: Performance on zero-shot labels (USPTO-5K-zeroshot dataset). BiEncoder combined with CrossEncoder achieves R@10 of 0.41 on approximately 3000 new labels.

example, the Bi-Encoder plays a more significant role in ranking the correct label into the top three, and when excluding the Cross-Encoder there is a large drop at NDCG@10, this behavior is expected as it is trained to rerank the top 30 predicted labels from the Bi-Encoder. For BM25 and GraphSage the performance drop at each level is about equal, however, for GraphSage the performance contribution is the least significant out of the four models.

Our second important observation is that in general removing pairs of models has a higher loss of performance, compared to the summation of losses when excluding the models individually, as observed in Figure 3. To investigate this result, we analyze the performance of each model decomposed on a subset of 600 classes from the USPTO-2M dataset. We do this by looking at the absolute difference between the pairs (BE, GS) and (CE, BM25) for each class, as can be viewed in Figure 4. Here we observe that both pairs have a high differential in class performance. This dichotomy in performance causes some uncertainty when the L2R model is only provided two models, which is caused by the label-agnostic nature of the aggregated ranking process. This effect is most noticeable when removing the Cross-Encoder and BM25, where the loss is most substantial. This loss is, however, exacerbated due to these two models contributing the most individually.

This issue is alleviated by the flexibility of our PARL model since it can easily integrate diverse component models into its process. So, it is important to have more than two models that provide different views, where the additional models can be viewed as tiebreakers in a voting mechanism, therefore lowering the uncertainty for the L2R model.

5.5. Embedding Visualization

Patent-label embeddings. Figure 5 shows the Bi-Encoder embeddings of patents and labels. We randomly sample five labels, *A22C*, *B41J*, *E04B*, *F15B*, and *G03G*, from the test set of USPTO-2M. The embeddings of these labels and the patents belonging to these labels are generated by the Bi-Encoder model trained on USPTO-2M, and then compressed to a 2-dimensional space using the t-SNE (Van der Maaten and Hinton, 2008) model. We can see that label embeddings are distributed closely to the embeddings of the patents of that type, indicating that the Bi-Encoder can represent semantically similar patents and labels close in the vector space, thus being able to retrieve correct labels for patents.

Patent-patent embeddings. Similarly, Figure 6 shows the GraphSAGE embeddings of patents from training and test sets. We use the same 5 labels as Figure 5, and for each label, we sampled patents from both the training and test sets. We find that patents from both training and testing for the same label type are distributed closely, indicating that the GraphSAGE embeddings are able to well capture patent-patent similarity.

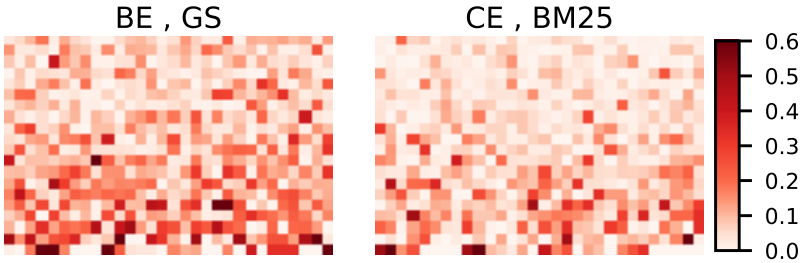

Figure 4: Absolute performance difference for 600 classes (sorted by frequency from top to bottom) between two component models for NDCG@5 and the USPTO-2M dataset. The difference between BE, GS (left) and between CE, BM25 (right).

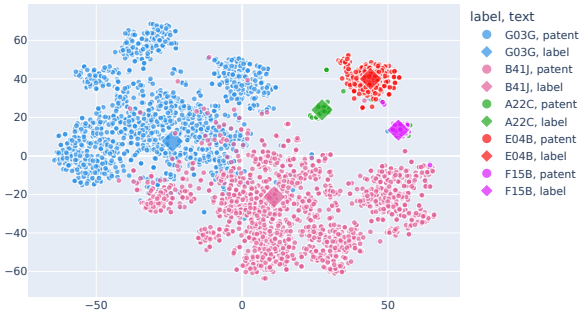

Figure 5: Embeddings of 5 labels and their associated patents. Each color represents one label type. The trained Bi-Encoder captures the close relationship between patents and their associated labels.

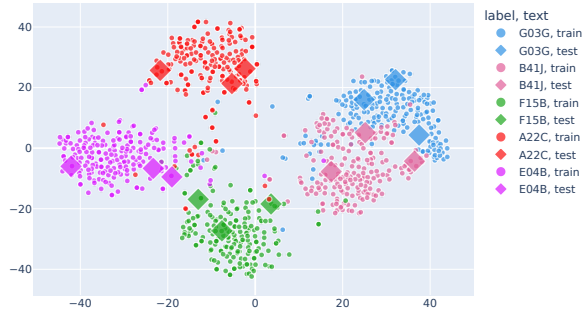

Figure 6: Embeddings of patents from both the training and test sets, categorized into 5 label types. Each label type is visually distinguished by a unique color. The GraphSAGE model demonstrates its ability to capture the close proximity of patents belonging to the same label.

6. Conclusion

In this work, we propose a multi-view ranking-based method for the task of multi-label patent classification. Our method consists of four ranking-based models that incorporate different views of patents and a meta-model that aggregates and re-ranks the candidate labels given by the four ranking models. The method is able to capture patent-label and patent-patent information, scale to large-scale labels, and adapt to new labels it has not seen before. Our method achieves a new state-of-the-art performance on public datasets.

7. Limitation

The datasets we constructed at the subgroup level only include the most frequently occurring top 1000, 5000, and 10,000 labels, which do not encompass the entire set of 300,000 labels. We encountered a lack of open-sourced data to augment all the patents in the USPTO-2M dataset with subgroup-level labels. Consequently, to fully evaluate the scalability of our model, the construction of a comprehensive subgroup-level dataset remains a task for future research.

The patents in both USPTO-2M and CLEF-IP have the title and abstract as their texts, without claims. The claim is to protect the inventors' rights without detailed technical information; the title, abstract, and claim are generally considered the most informative sections of a patent (Benzineb and Guyot, 2011). Similarly, a more comprehensive description of the labels is necessary. To fully harness the information present in a patent, we intend to augment the dataset with additional textual information. This will further introduce a new challenge of dealing with long texts for the representation modules in our method.

Another limitation of PARL is that it is label-agnostic, which can cause some uncertainty for the L2R model. This is partially alleviated by having four different views, each of which can act as a tiebreaker. However, to fully alleviate the confusion problem the aggregated ranking process should be label-aware. This improvement we also leave for future research.

8. References

Bashar Al-Shboul and Sung-Hyon Myaeng. 2011. Query phrase expansion using wikipedia in patent class search. In *Asia Information Retrieval Symposium*, pages 115–126. Springer.

Karim Benzineb and Jacques Guyot. 2011. Automated patent classification. In *Current challenges in patent information retrieval*, pages 239–261. Springer.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81.

Jia-Jia Cai, Jun Tang, Qing-Guo Chen, Yao Hu, Xiaobo Wang, and Sheng-Jun Huang. 2019. Multi-view active learning for video recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2053–2059. International Joint Conferences on Artificial Intelligence Organization.

Xiao-Lei Chu, Chao Ma, Jing Li, Bao-Liang Lu, Masao Utiyama, and Hitoshi Isahara. 2008. Large-scale patent classification with min-max modular support vector machines. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 3973–3980. IEEE.

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.

Qiang Cui, Shu Wu, Qiang Liu, Wen Zhong, and Liang Wang. 2018. Mv-rnn: A multi-view recurrent neural network for sequential recommendation.

Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144.

Lintao Fang, Le Zhang, Han Wu, Tong Xu, Ding Zhou, and Enhong Chen. 2021. Patent2vec: Multi-view representation learning on patent-graphs for patent classification. *World Wide Web*, 24(5):1791–1812.

Arousha Haghighian Roudsari, Jafar Afshar, Charles Lee, and Wookey Lee. 2020. Multi-label patent classification using attention-aware deep learning model. pages 558–559.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NIPS*.

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*.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 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.

Ralf Krestel, Renukswamy Chikkamath, Christoph Hewel, and Julian Risch. 2021. A survey on deep learning for patent analysis. *World Patent Information*, 65:102035.

Jieh-Sheng Lee and Jieh Hsiang. 2020. Patent classification by fine-tuning bert language model. *World Patent Information*, 61:101965.

Huahang Li, Shuangyin Li, Yuncheng Jiang, and Gansen Zhao. 2022. Copate: A novel contrastive learning framework for patent embeddings. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1104–1113.

Shaobo Li, Jie Hu, Yuxin Cui, and Jianjun Hu. 2018a. Deeppatent: patent classification with convolutional neural networks and word embedding. *Scientometrics*, 117:721 – 744.

Shaobo Li, Jie Hu, Yuxin Cui, and Jianjun Hu. 2018b. Deeppatent: patent classification with convolutional neural networks and word embedding. *Scientometrics*, 117(2):721–744.

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

Florina Piroi, Mihai Lupu, Allan Hanbury, and Veronika Zenz. 2011. Clef-ip 2011: Retrieval in the intellectual property domain. In *CLEF (notebook papers/labs/workshop)*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Arousha Haghighian Roudsari, Jafar Afshar, Wookey Lee, and Suan Lee. 2021. Patentnet: multi-label classification of patent documents using deep learning based language understanding. *Scientometrics*, 127:207 – 231.

Marco Seeland and Patrick Mäder. 2021. Multi-view classification with convolutional neural networks. *PLOS ONE*, 16:e0245230.

Pingjie Tang, Meng Jiang, Bryan (Ning) Xia, Jed W. Pitera, Jeffrey Welser, and Nitesh V. Chawla. 2020. Multi-label patent categorization with non-local attention-based graph convolutional network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:9024–9031.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

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.

Lizhong Xiao, Guangzhong Wang, and Yang Zuo. 2018. Research on patent text classification based on word2vec and lstm. In *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, volume 01, pages 71–74.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2018. Graph convolutional networks for text classification.

Jiong Zhang, Wei-Cheng Chang, Hsiang-Fu Yu, and Inderjit Dhillon. 2021. Fast multi-resolution transformer fine-tuning for extreme multi-label text classification. In *Advances in Neural Information Processing Systems*, volume 34, pages 7267–7280. Curran Associates, Inc.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense text retrieval based on pretrained language models: A survey.

You Zuo, Houda Mouzoun, Samir Ghamri Doudane, Kim Gerdes, and Benoît Sagot. 2022. Patent classification using extreme multi-label learning: A case study of french patents. In *SIGIR 2022-PatentSemTech workshop-3rd Workshop on Patent Text Mining and Semantic Technologies*.