A Label Informative Wide & Deep Classifier for Patents and Papers

Muyao Niu
Patsnap Ltd.
niumuyao@patsnap.com

Jie Cai
Patsnap Ltd.
caijie@patsnap.com

Abstract

In this paper, we provide a simple and effective baseline for classifying both patents and papers to the well-established Cooperative Patent Classification (CPC). We propose a label-informative classifier based on the Wide & Deep structure, where the Wide part encodes string-level similarities between texts and labels, and the Deep part captures semantic-level similarities via non-linear transformations. Our model trains on millions of patents, and transfers to papers by developing distant-supervised training set and domain-specific features. Extensive experiments show that our model achieves comparable performance to the state-of-the-art model used in industry on both patents and papers. The output of this work should facilitate the searching, granting and filing of innovative ideas for patent examiners, attorneys and researchers.

1 Introduction

Classifying patents and papers to a technology taxonomy is a crucial step to organize the massive knowledge and to discover innovative ideas. Patent examiners rely on the taxonomy to search for similar documents when granting or invalidating a patent application; attorneys use it to check whether the innovation points of an invention have been covered in previous literature; researchers use the taxonomy to monitor the technology trends in certain fields, and companies use it to outline the intellectual property landscape of its own or its competitors1. The most commonly used taxonomies are International Patent Classification (IPC) 1 and its newer version Cooperative Patent Classification (CPC) 2. Figure 1 illustrates the CPC hierarchy and the discriminative descriptions attached to each node.

1https://www.wipo.int/classifications/ipc/en/
2https://www.epo.org/searching-for-patents/helpful-resources/first-time-here/classification/cpc.html

Figure 1: Illustration of the CPC hierarchy.

The categorization is now mostly done manually by experts in patent offices. Due to the growing amount of patents and the limited number of domain experts, there has been an urge to automate the classification process. Also, as more and more technology innovations are published in patents, researchers with no background in patent classification may want to know which patents are most relevant to an academic paper. For this end, we aim to classify both patents and papers to the CPC subclass with more than 600 labels.

Classifying patents and papers to CPC subclass is a challenging task because (1) there are a large number of labels that covers almost all technology domains, and the differences between labels are often subtle; (2) although mass amount of labelled data for patents is available, the annotated data for paper-to-CPC is very limited. Since labelling papers with CPC labels requires expert knowledge, large-scale human annotation is very expensive.

In this paper, we leverage the CPC label descriptions and use the Wide-and-Deep network to integrate label information with semantic information from input texts. We also construct a distant-supervised dataset for papers. Our contributions are:

• We prove the effectiveness of the label features through the Wide-and-Deep structure in CPC classification on more than 600 labels.

• We achieve comparable performance to the
state-of-the-art on classifying both patents and papers to a widely-used technology taxonomy. Our model can serve as a simple and effective baseline for CPC classification tasks.

2 Related Work

2.1 Patent Classification

Most of the previous patent classification systems focus on developing more features derived from patent structures with traditional learning algorithms (Verberne and D’hondt, 2011). Going beyond document-level features, Cai and Hofmann (2007) and Qiu et al. (2011) capture the label hierarchy in model representations. It is flexible for traditional learning algorithms to integrate prior knowledge; however, it is difficult for them to generalize to unseen data.

Deep Neural Networks make efficient use of large-scale training data and generalize well to unseen data. There are a few NN-based patent classification methods. Grawe et al. (2017) uses Long Short Term Memory (LSTM) on 50 IPC subgroups and Li et al. (2018) applies Convolutional Neural Networks (CNN) to IPC subclasses classification.

There are some efforts on mapping papers to IPC in the context of patent retrieval combined with K nearest neighbours (KNN) classification. Cao et al. (2008) adopt a query-expansion approach to retrieve relevant patents and use a KNN classifier to label the research paper. Xiao et al. (2008) combine different scoring methods to rerank the retrieved IPC and achieved the best performance in the NTCIR-7 workshop for classify research papers to IPC system (Nanba et al., 2008). To our best knowledge, there has been no attempt tackling both tasks in one model.

2.2 Label Information

Leveraging label information is not new and is mostly accomplished by embedding labels and texts in the same space to measure their correlations (Yogatama et al., 2015; Zhang et al., 2018). Ma et al. (2016) adds prototypes to the label representation. Zhang et al. (2018) proposes to transform classification to a matching target between texts and labels for multi-task learning. Wang et al. (2018) further weights text features by the compatibilities between text and label embeddings via attention mechanism. Our model differs from previous studies in that we use the string-level similarity between label descriptions and input text instead of label embeddings. Based on the analysis of CPC classification system, we believe that string-level similarity can compensate for what semantic-level similarity cannot captures for patent classification tasks.

3 Model

3.1 The Label-text Feature

We discover that label descriptions can provide precise cues to classify a document which contains multiple semantic aspects. Figure 2 provide an example on the necessity of integrating label description. It should be noted that when patent examiners classify documents to the CPC system, they are also advised to use cue words and to search among label descriptions.

We integrate the label information through the label-text feature that captures the string-level relatedness between label descriptions and texts. Here we use BM25 score:

\[
BM25(D_k, x) = \sum_{i=1}^{n} \frac{idf(x_i) \cdot \frac{tf(x_i, D_k) \cdot (k_1 + 1)}{tf(x_i, D_k) + k_1 \cdot (1 - b + b \cdot \frac{|D_k|}{avgdl})}}{idf(x_i) \cdot \frac{tf(x_i, D_k) \cdot (k_1 + 1)}{tf(x_i, D_k) + k_1 \cdot (1 - b + b \cdot \frac{|D_k|}{avgdl})}}
\]

where \(x_i\) is the ith item in text \(x\); \(D_k\) is the label description for class \(k\); \(idf\) and \(tf\) are inverse document frequency and text frequency for \(x_i\) in \(D_k\) respectively; \(avgdl\) is the average description length; \(|D_k|\) is the document length; \(k_1\) and \(b\) are hyper parameters.

Figure 2: Example of the label description (left boxes) and classification cues (shadowed texts). The true label for the input text (right box) is B01D. Although the given text discussed neural networks, which is semantically closer to G06N, the classification cue of “ultrafiltration” decides for B01D eventually.

3.2 Learning to Classify

We learn to classify by using a neural network model that can take into account the label-text feature. The model consists of an encoder that takes in the text and the label-text feature and outputs a representation for each. This representation is then fed into a classifier that predicts the class label.

3.3 Model Training

The model is trained using the cross-entropy loss function.

\[
\text{Loss} = -\sum_{k} \sum_{x} y_{x,k} \log \hat{y}_{x,k}
\]

where \(y_{x,k}\) is the true label probability for document \(x\) and class \(k\); \(\hat{y}_{x,k}\) is the predicted probability for class \(k\).

3.4 Model Evaluation

We evaluate the model using the macro-averaged F1 score.

\[
F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]

where \(\text{Precision}\) and \(\text{Recall}\) are calculated for each class and then averaged over all classes.

3.5 Model Analysis

We analyze the model’s performance by looking at the confusion matrix and the precision-recall curve.

3.6 Model Limitations

The model may struggle with classifying documents that do not contain any label cues.

3.7 Future Work

We plan to extend this work to consider more complex label hierarchies and to incorporate more sophisticated attention mechanisms.

3.8 Conclusion

In this paper, we have presented a novel approach to patent classification that leverages label information. Our model shows promising results on both patent and research paper classification tasks.

References


The model outputs the probabilities for each of $L$ classes using sigmoid activation $\hat{y}_{deep} = \sigma(W^T_{deep}z_{deep} + b)$.

The Wide and Deep parts are concatenated at the top and are jointly trained through $\hat{y} = \sigma(W^T_{deep}z_{deep} + W^T_{wide}z_{wide} + b)$ in order to let the semantic and the string level relatedness complements each other when making the decision. In this way, the model simulates the behavior of patent examiners classifying a document: when they are uncertain which labels to assign (i.e. when semantic knowledge cannot provide a certain answer), examiners will resort to searching for cue words in the label descriptions for a clue.

4 Experimental Setup

4.1 Datasets

We remove stopwords and punctuations and choose the first 120 words per document. The word embeddings are 300 dimensional and initialized randomly. Kernel size for textCNN is 2,3,4 and 5, and the number of filters is 1024. For each CPC subclass, we use the descriptions of its own and of all its child labels. We train the model using Adam optimizer.

Datasets: Out of the USPTO patent set, We randomly sample 6.7 million abstracts as the patent training set and 60k as the testing set. For the paper testing set, the gold-standard is hard to obtain. We discover that some papers cited by patents are assigned CPC labels by European Patent Office, we collect those from the website \footnote{https://www.epo.org/index.html} and derive 4956 testing instances for paper-to-CPC classification. The datasets are described in table 1.

<table>
<thead>
<tr>
<th># instances</th>
<th># avg label per example</th>
</tr>
</thead>
<tbody>
<tr>
<td>training set</td>
<td>6.7 million</td>
</tr>
<tr>
<td>patents testing set</td>
<td>60k</td>
</tr>
<tr>
<td>papers testing set</td>
<td>4956</td>
</tr>
</tbody>
</table>

Table 1: Dataset Description

Evaluation metrics: As each patent/paper has one or more CPC labels, we measure our model from both the classification and the ranking perspectives with 3 metrics: (1) example-based precision/recall: the average precision/recall per instance. We measure precision and recall on the top1, top3 predictions and precision on all predictions with the probability score $\geq 0.5$. (2) macro precision/recall: the average precision/recall per class. (3) mean average precision (MAP): a
ranking-based metric that measures whether the right labels are placed before the wrong ones.

4.2 Classify Patents to CPC

We compare our WnD classifier on patents with two baselines: traditional textCNN and attention-textCNN. By comparing WnD with textCNN, we want to know whether the label-text feature can complement semantic information for classification; by comparing with attention-textCNN, we want to compare our label integration method with other label embedding-based methods. For the attention model, we borrowed the idea of label-embedding attentive model (Wang et al., 2018). The attention is a $T$-dimensional vector where $T$ is the text length. It calculates the importance of each word to the classification task.

The WnD achieves significant gains with label information (see table 2). It suggests that the complementary effects of string-level relatedness between label and texts indeed benefits the final classification decision. Our model also outperforms attention-textCNN. Although label embeddings are helpful for small label sets (around 10 labels) (Wang et al., 2018; Zhang et al., 2018), it is less effective on hundreds of labels. We suspect the reason is that the attention is not discriminative between classes. When the label set is large, many non-stop words may be important for classification. But their weights should vary for different classes, which can hardly be captured by the attention vector. Also, attention-textCNN has much more parameters than textCNN and tends to overfit on the training data.

The best reported numbers on patent-to-IPC in subclass level achieves 0.74 on precision (Verberne and D’hondt, 2011). Although can not be directly compared, our large-enough testing set makes it confident that we are comparable with the state-of-the-art system while being more scalable.

4.3 Classify Papers to CPC

There is not enough labelled data for the paper-to-CPC task. We can directly apply the model trained on patents to papers, but the performance will degrade significantly due to domain difference. For example, the word camera in papers is commonly referred to as photo capturing apparatus in patents. To deal with the domain-adaptation issue, we propose two approaches:

**distant supervision**: We auto-label papers using patent-paper citation. For each paper, we label it with the CPCs of the patents that cite it. We assume that papers cited by a patent should be relevant to the given patent in terms of background and technology domain. We get a total 1.7 million papers with abstract that are cited by the patents in the USPTO patent set, and each paper gets on average 2.8 labels. On these auto-labelled papers we then fine tune the model originally trained on patents.

**domain-adapted features**: The WnD structure enables us to incorporate domain-adapted features to the Wide part. Here we proposed two ways to add such features:

- **prototyping**: We pick the top 20 terms from the papers for each of the $K$ classes according to the tf-idf score. Those representative terms are used as the label descriptions for papers;

- **label expansion**: We train word embeddings using skip-gram (Mikolov et al., 2013) on papers and expand the original label descriptions with the 10 nearest words in the embedding space according to their cosine distance.

We compare our paper-to-CPC model with the best-performing KNN+reranking model (Xiao et al., 2008) introduced in section 2.1. We also want to compare our model with large-scale classification systems used in industry. In order to do that, we crawl from Google Patent the machine-classified CPC labels of scholar papers for our testing set $^5$, and we assume that the labels are ranked according to the order on the web page. Google classifies papers on the finest level. In order to compare our subclass results with it, we use only the subclass part of the first label, which is supposed to be the most confident one.

The comparison results are shown in table 3. WnD benefits from both transfer learning and domain-adapted features. Since the prototypes are automatically collected, it is possible to apply WnD to classification tasks where detailed label descriptions are not available.

Google Patent scores better on precision/recall@1, but performs less well on macro precision/recall. Since Google classifies the papers on a finer grained level, the classifier may receive more information during training, thus performing better on coarser grained levels. To

$^5$The data can be found by searching for a scholar paper in Google Patent. The machine-classified CPCs will appear in the information page.
Table 2: Test results on patents. For attention-CNN we used the implementation of attention mechanism from (Wang et al., 2018)

<table>
<thead>
<tr>
<th>Model</th>
<th>p@1</th>
<th>r@1</th>
<th>p@3</th>
<th>r@3</th>
<th>precision (prob &gt;0.5)</th>
<th>Macro p@1</th>
<th>Macro r@1</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>textCNN</td>
<td>75.51</td>
<td>55.24</td>
<td>40.64</td>
<td>78.92</td>
<td>62.16</td>
<td>63.77</td>
<td>32.15</td>
<td>75.68</td>
</tr>
<tr>
<td>attention-textCNN</td>
<td>70.63</td>
<td>51.49</td>
<td>38.11</td>
<td>74.17</td>
<td>51.69</td>
<td>56.98</td>
<td>26.15</td>
<td>72.46</td>
</tr>
<tr>
<td>WnD</td>
<td>77.11</td>
<td>56.42</td>
<td>41.18</td>
<td>80.12</td>
<td>64.01</td>
<td>67.81</td>
<td>34.18</td>
<td>76.60</td>
</tr>
</tbody>
</table>

Table 3: Test results on papers. textCNN and WnD are models trained on patents directly apply to papers. WnD+transfer refers to WnD fine tuned on auto-labelled papers.*The training set and granularity of Google Patent model may be different from other models. We put it here for the convenience to compare and discuss

investigate the effects of classification granularity on performance, we mapped the WnD subclass predictions to class level (128 label) and trained another WnD on class level. The p@1 and r@1 are 86.94% and 79.15% for WnD subclass-to-class and 82.43% and 74.72% for WnD class. The gap indicates the possible positive effects of fine-granularity training.

5 Conclusions and Future Works

In this paper, we propose a WnD classifier to map both patents and papers to CPC subclasses. The model captures both the string and semantic relatedness between labels and texts. We achieve comparative performance to the state-of-the-art models for both paper-to-CPC and patent-to-CPC tasks. We hope to contribute an intuitive, simple yet practically effective baseline for categorizing scientific publications.

Although CPC subclass already has over 600 labels, it is still a relative coarse granularity in the taxonomy. The finest level (subgroup) consists of over 200 thousand labels and provides much more detailed classification information. At the same time, with the explosion of labels, the task is much more challenging. In the future, we will go deeper into the taxonomy and try to explore the hierarchical relations between labels and improve the scalability of models for finer grained label sets.

6 Acknowledgements

We would like to thank Jian Yang for helping work through the codes and Yang Ding for his suggestions on the experiments. We would also like to thank the anonymous reviewers of EMNLP 2019; their comments and suggestions have helped us polish our arguments.

References


