# 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 competitors'. 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.

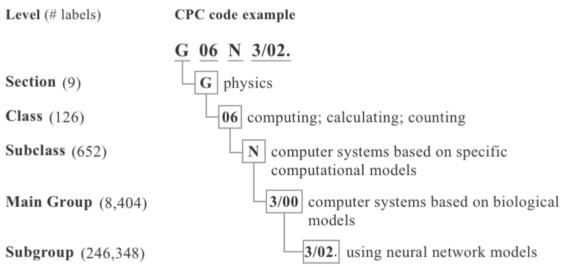
Level (# labels)
CPC code example
G 06 N 3/02.
Section (9)
G physics
Class (126)
06 computing; calculating; counting
Subclass (652)
N computer systems based on specific computational models
Main Group (8,404)
3/00 computer systems based on biological models
Subgroup (246,348)
3/02. using neural network models

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

[1] https://www.wipo.int/classifications/ipc/en/

[2] https://www.epo.org/searching-for-patents/helpful-resources/first-time-here/classification/cpc.html

*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing*, pages 3438–3443,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

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.

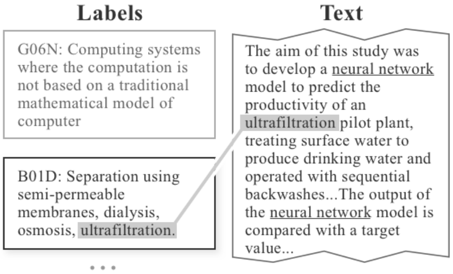
Labels
Text
G06N: Computing systems where the computation is not based on a traditional mathematical model of computer
B01D: Separation using semi-permeable membranes, dialysis, osmosis, ultrafiltration.
. . .
The aim of this study was to develop a neural network model to predict the productivity of an ultrafiltration pilot plant, treating surface water to produce drinking water and operated with sequential backwashes...The output of the neural network model is compared with a target value...

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 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 [3].

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:.

$$\mathrm{BM25}(\mathrm{D}_k, x) = \sum_{i=1}^{n} \mathrm{idf}(x_i) \frac{\mathrm{tf}(x_i, D_k)(k_1+1)}{\mathrm{tf}(x_i, D_k) + k_1(1-b+b\frac{|D_k|}{\mathrm{avgdl}})} \quad (1)$$

where $x_i$ is the $i$th item in text x; $\boldsymbol{D}_k$ 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.

[3] see IPC guidelines at https://www.wipo.int/export/sites/www/classifications/ipc/en/general/guidelines_where_to_classify.pdf

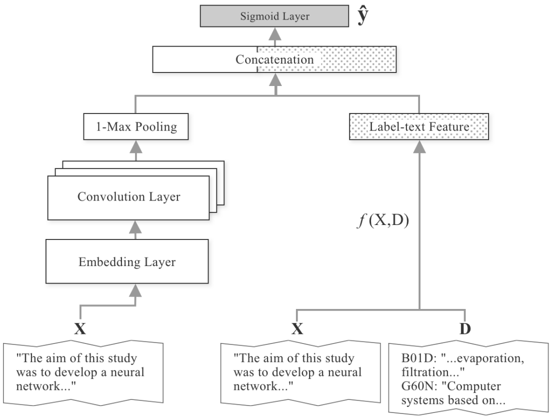
Sigmoid Layer
ŷ
Concatenation
1-Max Pooling
Label-text Feature
Convolution Layer
$f$(X,D)
Embedding Layer
X
X
D
"The aim of this study was to develop a neural network..."
"The aim of this study was to develop a neural network..."
B01D: "...evaporation, filtration..."
G60N: "Computer systems based on...

Figure 3: WnD classifier structure. The Deep part (left) captures semantic information; the Wide part (right) captures label-text relatedness. The label-text feature is a $K$-dimensional vector, where $K$ equals to the label set size.

### 3.2 Wide and Deep Structure

We adopt the Wide and Deep (WnD) neural network (Cheng et al., 2016) for text classification . Given a training set $(\boldsymbol{X}_n, \boldsymbol{D}, \boldsymbol{y}_n)_{n=1}^{N}$, where $\boldsymbol{X}_n$ is the input document texts, $\boldsymbol{D}$ is the input label descriptions and $\boldsymbol{y}_n$ is the true labels. The model outputs the probabilities for each of the $K$ classes $\hat{\boldsymbol{y}} \in R^K$. The training target is to minimize binary cross entropy loss: $L = \frac{1}{N}\frac{1}{K}\Sigma_{n=1}^{N}\Sigma_{k=1}^{K}\text{CE}(\boldsymbol{y}_n^{(k)}, \hat{\boldsymbol{y}}_n^{(k)})$, where $\boldsymbol{y}_n^{(k)}$ is the $k$th element in vector $\boldsymbol{y}_n$.

An overview of the WnD classifier is shown in figure 3. The model has two parts: the wide part and the deep part. The wide part takes in the label-text feature to capture string-level relatedness between the label descriptions and the text; the deep part maps the input text to word embeddings and go through a non-linear transformation to capture semantic-level relatedness between the text and the label.

**The Wide part**: The Wide part is a regression model with the form $\hat{\boldsymbol{y}}_{wide} = \sigma(\boldsymbol{W}_{wide}^T \boldsymbol{z}_{wide} + \boldsymbol{b})$, where $\boldsymbol{z}_{wide}$ is the label-text interaction features as described in section 3.1: $\boldsymbol{z}_{wide} = \text{BM25}(\boldsymbol{D}, \boldsymbol{X_n})$.

**The Deep part**: The Deep part is a non-linear transformation of the input text that aims to capture the semantic of the text. It can be a classic neural network for text encoding, such as RNN, CNN, or simple fully connected network. In this paper, we use textCNN (Kim, 2014) for the transformation because it is a simple baseline that works reasonably well. The Deep part transform the texts to a fixed-length representation $\boldsymbol{z}_{deep}$. The representation $\boldsymbol{z}_{deep}$ is then mapped to $K$ classes using sigmoid activation $\hat{\boldsymbol{y}}_{deep} = \sigma(\boldsymbol{W}_{deep}^T \boldsymbol{z}_{deep} + \boldsymbol{b})$.

The Wide and Deep parts are concatenated at the top and are jointly trained through $\hat{\boldsymbol{y}} = \sigma(\boldsymbol{W}_{deep}^T \boldsymbol{z}_{deep} + \boldsymbol{W}_{wide}^T \boldsymbol{z}_{wide} + \boldsymbol{b})$ 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 [4] and derive 4956 testing instances for paper-to-CPC classification. The datasets are described in table 1.

| | # instances | # avg label per example |
|---|---|---|
| training set | 6.7 million | 1.47 |
| patents testing set | 60k | 1.45 |
| papers testing set | 4956 | 1.30 |

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

[4] https://www.epo.org/index.html

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

| **Model** | **p@1** | **r@1** | **p@3** | **r@3** | **precision (prob>0.5)** | **Macro p@1** | **Macro r@1** | **MAP** |
|---|---|---|---|---|---|---|---|---|
| textCNN | 75.51 | 55.24 | 40.64 | 78.92 | 62.16 | 63.77 | 32.15 | 75.68 |
| attention-textCNN | 70.63 | 51.49 | 38.11 | 74.17 | 51.69 | 56.98 | 26.15 | 72.46 |
| WnD | **77.11** | **56.42** | **41.18** | **80.12** | **64.01** | **67.81** | **34.18** | **76.60** |

Table 2: Test results on patents. For attention-CNN we used the implementation of attention mechanism from (Wang et al., 2018)

| **Model** | **p@1** | **r@1** | **p@3** | **r@3** | **precision (prob >0.5)** | **Macro p@1** | **Macro r@1** | **MAP** |
|---|---|---|---|---|---|---|---|---|
| textCNN | 63.18 | 53.93 | 33.76 | 81.91 | 48.00 | 16.07 | 11.19 | 71.76 |
| WnD | 66.16 | 56.73 | 34.15 | 82.86 | 50.09 | 16.77 | 11.74 | 75.04 |
| KNN+reranking | 63.99 | 54.98 | 32.85 | 79.18 | NA | 14.70 | 10.60 | 72.37 |
| Google Patent* | **69.59** | **59.95** | NA | NA | NA | 14.28 | 10.25 | NA |
| WnD + transfer | 68.33 | 58.68 | 35.25 | 85.10 | **51.26** | 17.23 | 11.45 | 77.15 |
| WnD +transfer+Prototype | 68.94 | 59.12 | **35.60** | **85.87** | 51.11 | **17.63** | **12.39** | **77.84** |
| WnD +transfer+labelExpand | 68.69 | 58.96 | 35.31 | 85.22 | 50.37 | 17.16 | 11.71 | 77.53 |

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

Lijuan Cai and Thomas Hofmann. 2007. Exploiting known taxonomies in learning overlapping concepts. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 714–719. Morgan Kaufmann Publishers Inc.

Guihong Cao, Jianyun Nie, and Lixin Shi. 2008. Ntcir-7 patent mining experiments at rali. pages 347 – 350. Proceedings of NTCIR-7 Workshop Meeting.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10. ACM.

Mattyws F. Grawe, Claudia A. Martins, and Andreia Gentil Bonfante. 2017. Automated patent classification using word embedding. In *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18-21, 2017*, pages 408–411. IEEE.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014*

*Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

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

Yukun Ma, Erik Cambria, and Sa Gao. 2016. Label embedding for zero-shot fine-grained named entity typing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 171–180.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Hidetsugu Nanba, Atsushi Fujii, Makoto Iwayama, and Taiichi Hashimoto. 2008. Overview of the patent mining task at the ntcir-7 workshop. pages 293 – 298. Proceedings of NTCIR-7 Workshop Meeting.

Xipeng Qiu, Xuanjing Huang, Zhao Liu, and Jinlong Zhou. 2011. Hierarchical text classification with latent concepts. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 598–602. The Association for Computer Linguistics.

Suzan Verberne and Eva D'hondt. 2011. Patent classification experiments with the linguistic classification system LCS in CLEF-IP 2011. In *CLEF 2011 Labs and Workshop, Notebook Papers, 19-22 September 2011, Amsterdam, The Netherlands*, volume 1177 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2321–2331.

Tong Xiao, Feifei Cao, Tianning Li, Guolong Song, Ke Zhou, Jingbo Zhu, and Huizhen Wang. 2008. KNN and re-ranking models for english patent mining at NTCIR-7. In *Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, NTCIR-7, National Center of Sciences, Tokyo, Japan, December 16-19, 2008*. National Institute of Informatics (NII).

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics.

Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2018. Multi-task label embedding for text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4545–4553. Association for Computational Linguistics.