# Leveraging Label Semantics and Correlations for Judgment Prediction

Yu Fan, Lei Zhang(✉), and Pengfei Wang

School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China
fanyubupt@gmail.com, {zlei,wangpengfei}@bupt.edu.cn

**Abstract.** Automatic judgment prediction is a classic problem in legal intelligence, which aims to predict the relevant violated articles based on the fact descriptions. Generally, both semantics and relations of articles are valuable information to solve this problem. However, previous work usually threats this problem as a classification task while these two types of information are not well explored, which makes previously proposed methods less effective. In this paper, we design a novel ***G***raph-Based ***La***bel ***M***atching Network (GLAM for short) to address this issue. Specifically, GLAM first builds a heterogeneous graph to capture both semantics and correlations among articles. Based on this, a graph convolutional network is then utilized to learn robust article representations. Finally, a matching model is applied between article representations and fact representations to generate the matching score for judgment prediction. Experimental results on two real-world judicial datasets demonstrate that our model has more significant effect on judgement prediction than the state-of-the-art methods.

**Keywords:** Multi-label classification · Judgment prediction · Graph-Based Label Matching Network

## 1 Introduction

The task of judgment prediction aims to determine the relevant violated articles based on the fact descriptions of criminal cases. It plays an important role in legal assistant system which can provide a handy reference for legal experts and improve their working efficiency [26]. Generally, the task is regarded as a multi-label classification problem. When making predictions based on descriptions, we usually introduce articles information (e.g., semantics and relations) to improve accuracy [9,13,16]. However, consider the complexity of the judicial system, there still exists two problems that have not been completely solved or even ignored, which makes existing methods less effective, as described below:

**Negligence of the Fact Relations Among Articles.** Usually, the number of articles corresponding to different fact descriptions is dynamic which makes the prediction more difficult. However, previous work has not considered that


Z. Dou et al. (Eds.): CCIR 2020, LNCS 12285, pp. 70–82, 2020.
https://doi.org/10.1007/978-3-030-56725-5_6

some articles have a high probability of co-occurrence regularly based on the facts. For example, in the scenario which is similar to Table 1, intentional injury and intentional destruction of property crime have a high probability of being violated simultaneously. We define the article co-occurrence as the fact relations among articles. The fact relations which have been ignored by previous works have a great impact on judgment prediction.

**Confusing Articles.** There are a bunch of confusing article pairs. The definitions of them only differ in a specific act (e.g., theft and robbery) and the circumstances in corresponding cases are usually similar with each other, which lead to confusion in classification. Previous work introduced article semantic information into classification but cannot fully reveal the confusing semantic relation. Hu et al. [9] introduced several discriminative attributes but it can only solve the confusing pairs what he proposed. Others use compressed or extract label semantic information to help classify which can't fully capture the semantic relations [16,22].

Besides, previous work applied traditional deep learning models such as convolutional neural networks [11] and long short-term memory [7] to express articles, which can capture semantic and syntactic information in local consecutive word sequences well, but may ignore global word co-occurrence in a corpus which carries non-consecutive and long-distance semantics [17].

**Table 1.** An example of the judgment case, including a fact and two articles violated.

| | |
|---|---|
| **Fact** | At 18 o'clock on August 7th, 2013, Song and Chen had a dispute, then Song held a steel pipe to beat Chen Mou, and used a steel pipe to fight the refrigerator, computer display and other property in Chen's shop... |
| **Article** | **Article 234**: Crime of intentional injury. Those who intentionally injure other people shall be sentenced to...<br>**Article 275**: Crime of intentional destruction of properties. Deliberately destroying public and private |

To solve the problems raised above, in this paper, we propose a novel ***G***raph-Based ***La***bel ***M***atching Network (GLAM for short). We find that article relations can be more fully represented by graphs. Therefore, we introduce the graph structure for article expression.

We define articles and words in articles as nodes, and we introduce multi-relation (e.g., article-article relation, article-word relation) as edges of the graph.

Then we put the heterogeneous graph into a graph convolutional network to express article labels. Fact and label representations will be put into a matching model with co-attention mechanism to generate the affinity matrices. The final matching score is produced by aggregating affinity matrices between articles and fact for judgment.

For the purpose of evaluating the performance of our proposed model, we conduct experiments on two legal datasets. Experimental results demonstrate that our approach significantly outperforms other state-of-the-art models. We also designed several sub-experiments to verify the superiority of our structured label graph.

## 2 Related Work

In this section we provide a brief overview on the following three related research areas.

**Judgment Prediction.** Automatic judgment prediction is a typical task in legal intelligence. Generally, this task will be cast as a text classification problem. Researchers usually extract effective features from text and apply machine learning methods to make judgments [1]. Hu et al. [9] introduced discriminative attributes to enhance the connections between the fact descriptions and charges, and these attributes and charges are inferred simultaneously. Then researchers incorporate attention mechanisms for articles and facts. For example, Luo et al. [16] proposed an attention-based neural model for charge prediction by incorporating the relevant articles. Long et al. [15] utilized the attention mechanism to model the complex semantic relations among facts, pleas, and articles. Wang et al. [22] introduced unified dynamic pairwise attention model for classifications over articles. In their work, a pairwise attention model based on article definitions is incorporated into the classification model to help alleviate the case imbalance problem and confusing charge classification problem. But these methods do not consider to leverage article information adequately.

**Graph Convolutional Networks.** Graph Convolutional Networks (GCN) approaches fall into spectral-based and spatial-based [23]. Among them, the application of spectral-based method is more extensive currently. Henaff et al. [6] proposed a strategy to learn the graph structure from the data and applied the model to image recognition, text categorization. Bruna et al. [2] proposed the first spectral convolutional neural network on graphs. Defferrard et al. [4] optimized spectral GCN by defining a filter as Chebyshev polynomials of the diagonal matrix of eigenvalues. Kipf and Welling [12] simplified the original frameworks to improve scalability and classification performance in large-scale networks. GCN is applied to deal with structured datasets, so a number of papers viewed a document or a sentence as a graph of word nodes for text classification [2,4,6,12,17]. Yao et al. [24] regarded the documents and words as nodes and construct the corpus graph.

**Semantic Matching.** The semantic matching is usually applied to learn the similarity information. Generally, they create a matching matrix which is well for scenarios like question answering [14], natural language inference, and information retrieval [10], etc. Hu et al. [8] firstly generates local matching patterns and composites them by multiple convolution layers to produce the matching score. Shen et al. [20] utilized the word level similarity matrix to discover fine-grained alignment of two sentences. Guo introduced a novel retrieval model by viewing the match between queries and documents as a transportation problem [5]. Wan et al. [21] applied Bi-LSTM to sentences and introduced interaction tensor to calculate the match between sentences. This technique benefits model to learn the semantics with richer representations and then perform matching with these representations. In our work, we use labels representation from GCN and facts pair for semantic matching, by this we formalize the traditional crime classification task into a matching task.

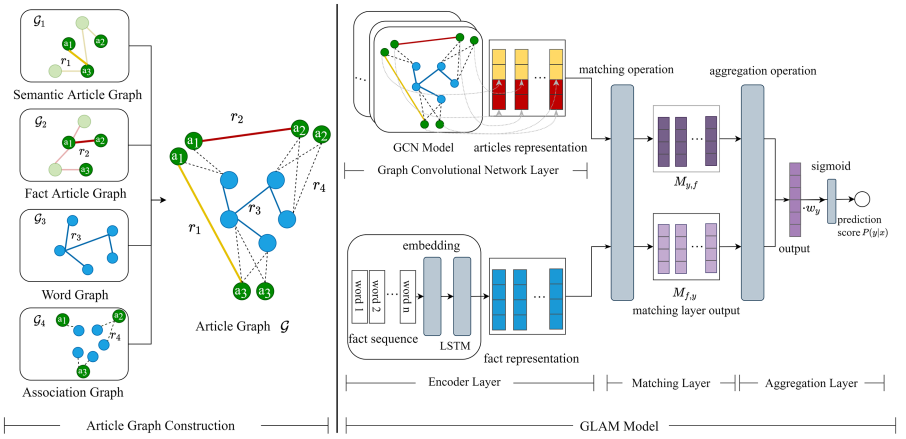


**Fig. 1.** On the left is the article graph construction process, on the right is the overall architecture of Graph-Based Label Matching Network (GLAM).

## 3 Method

In this section, we start with the problem formalization of judgement prediction. We then describe the construction of article graph $\mathcal{G}$. Based on these we introduce our GLAM model in detail. We finally present the learning and prediction procedure of GLAM.

### 3.1 Formalization

In judgment classification, let $X = \{x_1, x_2, \ldots, x_{|X|}\}$ denote all the facts, $\mathcal{A} = \{a_1, a_2, \ldots, a_{|A|}\}$ denote all the articles, $Y = \{y_1, y_2, \ldots, y_{|Y|}\}$ denote the set of all possible label concepts where each $y_i \in (0, 1)$ indicates whether article $a_i$ is violated or not. $|X|$ and $|Y| = |A|$ represent the total number of facts and labels. Each instance is represented as a tuple $(x_k, Y_k)$, where $x_k \in X$ represents the k-th fact, $Y_k \subseteq Y$ represents the article set assigned to $x_k$.

Given a fact $x$ and the article set $\mathcal{A}$, we aim to generate a relevance score for each label $y$ to check whether they are relevant or not.

### 3.2 Article Graph Construction

In this study, we construct an undirected heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ to formulate different information of articles where $\mathcal{V}$, $\mathcal{E}$, $\mathcal{R}$ are the set of nodes, edges and relations respectively. As Fig. 1 shows, $\mathcal{G}$ can be divided into four sub-graphs $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3 \cup \mathcal{G}_4$, which represent four types of relations correspondingly: (1) an article graph $\mathcal{G}_1$ modeling semantic relations (2) an article graph $\mathcal{G}_2$ modeling fact relations (3) a word graph $\mathcal{G}_3$ indicating the word co-occurrence between pairs of words, and (4) an association graph $\mathcal{G}_4$ involving association relations between articles and their words.

**Semantic Article Graph** $\mathcal{G}_1$. This graph represents the semantic relation between articles. It is denoted as $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1, \mathcal{R}_1)$. In particular, the set of nodes $\mathcal{V}_1 = \mathcal{A} = \{a_1, a_2 ... a_{|A|}\}$ includes all of the articles, and $\mathcal{R}_1$ is the semantic relation. We define $s(a)$ as the set of words in article $a$, so the semantic similarity which can be taken as the weight of edge between article $a_i$ and $a_j$ is written as follows:

$$g(a_i, a_j) = \frac{count(s(a_i) \bigcap s(a_j))}{count(s(a_i) \bigcup s(a_j))} \tag{1}$$

where the numerator is the number of common words in two articles, the denominator is the number of all words in them.

**Fact Article Graph** $\mathcal{G}_2$. This graph represents the fact relation between articles which is denoted as $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2, \mathcal{R}_2)$. The set of nodes $\mathcal{V}_2 = \mathcal{A}$ includes all of the articles, and $\mathcal{R}_2$ is the fact relation. For the edge of article pair $(a_i, a_j)$, the correlation weight is computed by point-wise mutual information (PMI) [3] as follows:

$$\text{PMI}(a_i, a_j) = log \frac{p(a_i, a_j)}{p(a_i) \times p(a_j)} \tag{2}$$

where $p(a_i, a_j)$ is the probability of article $a_i$ and $a_j$ are violated in one fact which is calculated by dividing the number of occurrences by the total number, and $p(a_i)$ is the probability of article $a_i$ is violated in one fact. Considering that our graph has no negative weights, this edge will not exist when the $PMI(a_i, a_j)$ is less than 0.

**Word Graph** $\mathcal{G}_3$. This graph represents the word co-occurrence relation between words which is denoted as $\mathcal{G}_3 = (\mathcal{V}_3, \mathcal{E}_3, \mathcal{R}_3)$. The set of nodes $\mathcal{V}_3 = W$ which represents all of the words in articles. The edges between words are built by word co-occurrence in the whole corpus. We still choose PMI measure to calculate the weight between two words.

**Association Graph** $\mathcal{G}_4$. This graph formulates the connection among the articles and their words, we define it as $\mathcal{G}_4 = (\mathcal{V}_4, \mathcal{E}_4, \mathcal{R}_4)$. The set of nodes $\mathcal{V}_4 = \{A \bigcup W\}$. The edge between the tuple of article and word are built by word occurrence in articles. The weight of edge is term frequency-inverse document frequency (TF-IDF) of the word in the article. Obviously, applying TF-IDF as weight is better than applying term frequency because TF-IDF tends to give high weights for words that are important to this article, and low weights for words that are common to all articles.

### 3.3 GLAM Model

This section describes our Graph-Based Label Matching Network (GLAM) in detail. Figure 1 shows the architecture of GLAM model.

**Graph Convolutional Network Layer.** From the above, we build a heterogeneous graph $\mathcal{G}$ with the correlation matrix $A$ where each element is calculated in the method defined above. The weight of edge will be zero when there is no edge between two nodes. Besides, we define a feature matrix $X$ containing semantic features about words and articles. Then we apply a GCN model to

generate new article representations. Considering efficiency and effectiveness, we take a 2-layer GCN with randomly initialized weights:

$$Z = \hat{A}\ ReLU(\hat{A}XW^{(0)})W^{(1)} \tag{3}$$

where $Z \in R^{n \times k}$ is the output matrix, $W^0$ and $W^1$ are parameters need to learn, and $k$ is the dimension of output features. And $\hat{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ where $D$ is the degree matrix of graph $G$. As Fig. 1 shows, two types of article nodes output two types of representations. We concatenate two types of representations as the final article representations which can be formulated as $V_Y \in R^{|Y| \times 2k}$.

**Encoder Layer.** In this section, we will design a encoder to generate fact representations. In juridical field, each fact is described by a set of words. The encoder encodes the discrete input sequence into continuous hidden states. More formally, we define $\mathbf{V} = \{\mathbf{v}_i \in \mathbb{R}^d | i = 1, 2, \dots\}$ denote all the word vectors for each fact in a D-dimensional continuous space. Given each fact $x$, we aggregate the word vectors to obtain its semantic matrix $\mathbf{V}_f$ as $[\mathbf{h}_f(1), ..., \mathbf{h}_f(n)]$, where $n$ is the fixed length of $\mathbf{V}_f$, and $\mathbf{h}_f(t)$ are regarded as the representation at time step t, which are obtained by LSTM [7]:

$$\mathbf{h}_f(t) = LSTM(\mathbf{v}_t : t \in x, \mathbf{v}_t \in \mathbf{V}, \mathbf{h}_f(t-1)) \tag{4}$$

**Matching Layer.** Matching layer is dedicated to select attentive semantics from both facts and labels to generate relevance scores for matching. We have contextual vector representations of the context $V_f \in R^{n \times h}$ as fact representations from encoder layer. And we get the expression of labels (i.e., articles) $V_Y \in R^{|Y| \times 2k}$ where $h = 2k$ from GCN layer. Then we propose a co-attention mechanism to compute the affinity matrix between facts and labels:

$$\begin{aligned} M_{f,y} &= s(V_Y V_f^T)V_f \\ M_{y,f} &= s((V_Y V_f^T)^T)V_Y \end{aligned} \tag{5}$$

where $s(\cdot)$ is the softmax function to the second dimension. $M_{f,y}$ and $M_{y,f}$ are fact-to-articles and articles-to-fact affinity matrices respectively.

**Aggregation Layer.** In aggregation layer, we will integrate attention context between articles and fact to obtain the matching score. We employ the sigmoid function $\sigma(\cdot)$ to output the relevance score of $(x, y)$ through Eq. (6)

$$P(y|x) = \sigma(\mathbf{w}_y \cdot [g(\mathbf{M}_{f,y}); g(\mathbf{M}_{y,f})]) \tag{6}$$

where $\mathbf{w}_y$ is parameter need to learn, and $g(\cdot)$ aggregates one matrix by columns into a single vector.

### 3.4 Learning and Prediction

Finally, by considering all facts and their label sets, we obtain our learning approach as follows:

$$\mathcal{L}(x, Y_x) = \sum_{x \in X} \Big( \sum_{y \in Y_x} \big( \ln P(y, x) - \sum_{\bar{y} \in C - Y_x} \ln P(\bar{y}, x) \big) \Big) \tag{7}$$

where $\bar{y}$ is each negative label mined from siblings of $y$. We use the Adam optimizer and update the parameters of our model for each iteration according to Eq.(7).

To summarize, the matching strategy can be described as: for all article labels, we first generate their representations through GCN model. Based on the fixed label representations, given a fact $x$, the best label set is a combination of assignments with the highest score from each label given the input:

$$O_y(x, y) = \sum_{y \in C} I(P(y, x) > \delta_y) \tag{8}$$

where $I(\cdot)$ denotes the indicator function, $O_y(x, y)$ is the relevance score function when feeding label set $y$ to $x$, and $\delta_y$ is the learned threshold of label $y$.

According to Eq. (6) and Eq. (7), for each fact, we only need to conduct a forward computation to generate the scores for each label.

**Table 2.** Statistics of the two legal datasets for experiments.

| Dataset | #Fact | #Articles | Average fact description size | Average article definition size | Average law set size per fact | Average article set size per fact |
|---|---|---|---|---|---|---|
| Fraud | 17,160 | 70 | 1,455 | 136 | 2.6 | 4.3 |
| CAIL | 204,231 | 183 | 1,444 | 129 | 1.4 | 1.3 |

## 4 Experiment

In this section, we evaluate GLAM by comparing with several state-of-the-art methods. We first introduce the experimental settings, then we analyze the experimental results on the judgment prediction task.

### 4.1 Dataset and Experimental Setup

This section describes the dataset and experimental settings of our work.

**Dataset.** We ran our experiments on two real-world legal datasets which are Fraud dataset and CAIL dataset respectively.

– **Fraud** [22] comprises 17,160 criminal cases related with fraud. These data are crawled from China Judgment Online[1] and span from Jan. 2016 to June. 2016.

[1] http://wenshu.court.gov.cn/.

– **CAIL**[26] a public dataset from Chinese AI and Law challenge (CAIL2018). The cases in the dataset contain two parts, i.e., fact description and corresponding judgment result (including laws, articles, and charges).

We extract fact descriptions and applicable articles from datasets. After preprocessing we obtain 17,160 facts on the Fraud dataset, and 204,231 facts on the CAIL dataset. The detailed statistics of two datasets are shown in are shown in Table 2. Finally, we split all the datasets into two non-overlapping parts, the training set and testing set, with a ratio 8:2 and we randomly selected 10% of training set as validation set.

**Parameter Settings.** For the baselines, to make a fair comparison, we follow the reported optimal parameter settings and optimize them using the validation set. We implement our method in Pytorch. In GCN layer, we use 300-dimensional GloVe [18] word embeddings as the word feature. We set the batch size as 64, embedding dim of fact encoder as 300, the dimension of output feature in GCN as 128, and the hidden size of encoder layer as 256. We use Adam optimizer which is determined from 0.1 to 0.0001. The initial learning rate is 0.0015 with 0.9 exponential decay. For each fact description, we set maximum number of words is 500.

**Evaluation Metric.** We adopt Jaccard, macro precision (Macro-P), macro recall (Macro-R) and macro F-measure (Macro-F) which are widely used in the classification task to evaluate the performance. Differences are considered statistically significant when the p–value is lower than 0.05.

### 4.2 Baselines

To evaluate the performance of our methods, we compare our model[2] with the following methods:

– **BP-MLL**: It is derived from the popular backpropagation algorithm that captures the characteristics of multi-label learning by replacing its error function with a defined new error function [25].
– **CC**: Classifier Chains [19] is a binary association method for multi-label classification, thinking that each label is an independent binary problem.
– **TextCNN-MLL**: A convolutional neural network [11] which denotes multiple filter widths as text classifier, and employs a new error function similar to BP-MLL.
– **TOPJUDGE**: A topological multi-task learning framework for judgment prediction [26], which applies multiple subtasks and DAG dependencies to judgment prediction.
– **DPAM**: A unified Dynamic Pairwise Attention Model [22] that fusing article semantics into a pairwise attention matrix for judgment prediction. We use sequential form of DAG to model the dependencies between laws and articles.

CC and TextCNN-MLL were using Scikit−multilearn. For DPAM, BP-MLL, and TOPJUDGE, we use the code released by their authors.

[2] https://github.com/IntelligentLaw/GLAM.

**Table 3.** Performance on judgment prediction between the baselines and GLAM (all the values in the table are percentage numbers with% omitted). The best performance in each case is written in bold.

| Dataset | Fraud | | | | CAIL | | | |
|---|---|---|---|---|---|---|---|---|
| Metrics | Macro-P | Macro-R | Macro-F | Jaccard | Macro-P | Macro-R | Macro-F | Jaccard |
| BP-MLL | 45.1 | 30.4 | 34.4 | 60.1 | 41.6 | 30.2 | 33.6 | 59.7 |
| CC | 43.2 | 28.6 | 33.6 | 58.5 | 42.1 | 32.5 | 35.6 | 62.6 |
| TextCNN-MLL | 68.5 | 34.3 | 40.5 | 65.5 | 76.3 | 54.3 | 60.1 | 72.3 |
| TOPJUDGE | 68.9 | 35.1 | 40.7 | 65.8 | 77.1 | 54.9 | 61.1 | 72.9 |
| DPAM | 71.2 | 35.5 | 43.5 | 67.9 | 78.3 | 57.7 | 63.3 | 74.9 |
| GLAM | **71.5** | **46.5** | **52.8** | **75.2** | **81.1** | **68.1** | **71.5** | **81.5** |

### 4.3 Comparison Against Baselines

We compare GLAM to the state-of-the-art baseline methods for judgment prediction. The experimental results on the two datasets are shown in Table 3. The results show obviously that our model achieves the best performance on all metrics.

Compared with DPAM which performs best among baselines, we can infer that our model has highly improved on Macro-R (10.4% in CAIL and 10.0% in Fraud) and slightly improved in Macro-P (2.8% in CAIL and 0.3% in Fraud). This phenomenon might indicates that for each fact, the accuracy of predicting its positive label sets (violated articles) has been greatly improved, and the probability of misjudging irrelevant articles decreases slightly. DPAM considers semantic interactions between each pair of articles which does not adequately leverages information of articles. That is the reason why GLAM performs better than DPAM.

In other methods of baselines, the shallow model BP-MLL and CC performs worst. Comparing two models, CC performs well on Fraud dataset but not as good as BP-MLL on another dataset. The reason is that CC mechanism is flawed: if CC misclassifies a label, the incorrect label is passed on to the next classifier and sway the next classifier to a wrong decision [19]. The other three deep neural models in baselines performs better than shallow model. The result of TextCNN-MLL represents the ability of deep neural networks to learn representations more powerfully than shallow model. TOPJUDGE take the topological properties of multi-task into consideration but have a lower performance than DPAM. The reason is that both models use a multitasking learning framework, but DPAM introduces a pairwise attention mechanism based on article definitions to alleviate the label imbalance problem. In conclusion, GLAM achieves promising improvements which indicates the effectiveness of our model.

### 4.4 Analysis on the Graph of Articles

GLAM design a graph containing semantics and multi-relation among articles (e.g., semantic relations and fact relations). Then we put the heterogeneous

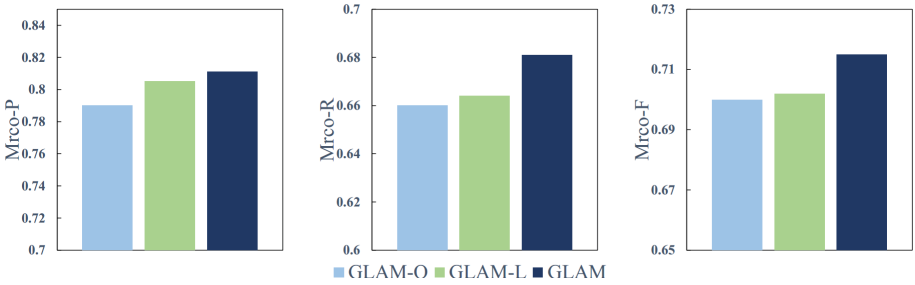


**Fig. 2.** Performance comparison of the GLAM model with its two sub-variant models GLAM-O and GLAM-L on CAIL dataset in terms of Marco-P, Macro-R and Macro-F1.

**Table 4.** Performance on judgment prediction between GLAM and GLAM-L (all the values in the table are percentage numbers with% omitted)

| Dataset | Fraud | | | | CAIL | | | |
|---|---|---|---|---|---|---|---|---|
| Metrics | Macro-P | Macro-R | Macro-F | Jaccard | Macro-P | Macro-R | Macro-F | Jaccard |
| GLAM-L | 70.7 | 44.5 | 50.9 | 74.7 | 80.5 | 66.4 | 70.2 | 81.0 |
| GLAM | **71.5** | **46.5** | **52.8** | **75.2** | **81.1** | **68.1** | **71.5** | **81.5** |

graph into a graph convolutional network. In this section we conducted experiments to verify that the various article information we introduced has worked.

Firstly, we delete the edges that represent fact relations among articles from our model and name it GLAM-L. The performance comparison between GLAM and GLAM-L is shown in Table 4. We can observe that GLAM gets the greatest improvement in Macro-R (2.0% in Fraud and 1.7% in CAIL). This proves that introducing fact relations in articles can increase the proportion of positive cases predictions. The performance improvement of GLAM on Macro-P metric is slight (0.8% and 0.6%), it indicates that the factual relation is less effective when predicting irrelevant articles. Correspondingly, when judging irrelevant articles, semantic information (other parts of the graph) have a greater effect.

Then, we delete all edges between articles which means remove the article relations from GLAM. We get a model that only contains word occurrence in articles and word co-occurrence information [24], which is named GLAM-O. We further compare the two sub-models GLAM-O and GLAM-L as well as GLAM to show their different effectiveness. From the result which shows in Fig. 2 we have the following observation: (1) A graph containing only word co-occurrence information and word occurrence in articles information can give a relatively good result, but not as good as the other two. (2) Adding semantic relations between articles on the basis of GLAM-O has slightly improved the performance, we can regard it as a supplement to the semantic information. (3) GLAM performs best on all metrics which verifies the significance of considering both semantics and two types of label relations for judgment prediction.

When comparing Table 3 and Table 4, it is worth mentioning that the experimental results of GLAM-L and DPAM on Macro-P are different in two datasets. On CAIL dataset, GLAM-L achieve a significant improvement on Macro-P compared with DPAM. But on Fraud dataset, GLAM-L has reduced performance on Macro-P by around 0.5% compared with DPAM. This is because the articles contained in the CAIL dataset are parallel (e.g., theft and robbery), it is more important to distinguish the differences of the keywords between them. But the Fraud dataset contains definitions of many concepts in judgment such as the concept of legitimate defense and joint crime which need more contextual information for classification. However, the application of the label relations in facts can effectively make up for this deficiency.

## 5 Conclusion

Judgment prediction is a crucial task in legal intelligence which leverages label information inadequately. We emphasize the importance of label semantics and relations, then we define label correlations and introduce the graph neural network to construct the label information. Moreover, we encode facts and put facts and labels representations into a matching model with co-attention mechanism to generate a relevance score for judgment. The experimental results show that GLAM achieves outperforms baseline methods and the information we introduced improves the judgment prediction.

In the future, we will explore from the following two aspects: (1) we will further analyze the significance of articles information to judgment prediction. (2) we will apply our model to other complex multi-task text classification problems.

**Acknowledgments.** This research work was partially supported by the National Natural Science Foundation of China under Grant No. 61802029, and Open Project Funding of CAS-NDST Lab under Grant No. CASNDST202005.

## References

1. Aletras, N., Tsarapatsanis, D., Preoţiucpietro, D., Lampos, V.: Predicting judicial decisions of the european court of human rights: a natural language processing perspective. Peer J **2** (2016)
2. Bruna, J., Zaremba, W., Szlam, A., Lecun, Y.: Spectral networks and locally connected networks on graphs. In: International Conference on Learning Representations (ICLR) (2013)
3. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Comput. Linguist. **16**(1), 22–29 (1990)
4. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Neural Information Processing Systems, pp. 3844–3852 (2016)

5. Guo, J., Fan, Y., Ai, Q., Croft, W.B.: Semantic matching by non-linear word transportation for information retrieval. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 701–710 (2016)
6. Henaff, M., Bruna, J., Lecun, Y.: Deep convolutional networks on graph-structured data (2015)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
8. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Neural Information Processing Systems (NIPS), pp. 2042–2050 (2014)
9. Hu, Z., Li, X., Tu, C., Liu, Z., Sun, M.: Few-shot charge prediction with discriminative legal attributes. In: Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, 20–26 August 2018 (2018)
10. Huang, P., He, X., Gao, J., Deng, L., Acero, A., Heck, L.P.: Learning deep structured semantic models for web search using clickthrough data, pp. 2333–2338 (2013)
11. Kim, Y.: Convolutional neural networks for sentence classification. In: Empirical Methods in Natural Language Processing, pp. 1746–1751 (2014)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2016)
13. Lauderdale, B.E., Clark, T.S.: The supreme court's many median justices. Am. Polit. Sci. Rev. **106**(04), 847–866 (2012)
14. Lin, J.J.: An exploration of the principles underlying redundancy-based factoid question answering. ACM Trans. Inf. Syst. **25**(2), 6 (2007)
15. Long, S., Tu, C., Liu, Z., Sun, M.: Automatic judgment prediction via legal reading comprehension. CoRR abs/1809.06537 (2018)
16. Luo, B., Feng, Y., Xu, J., Zhang, X., Zhao, D.: Learning to predict charges for criminal cases with legal basis. CoRR abs/1707.09168 (2017)
17. Peng, H., et al.: Large-scale hierarchical text classification with recursively regularized deep graph-CNN. In: The Web Conference, pp. 1063–1072 (2018)
18. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
19. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Mach. Learn. **85**(3), 333–359 (2011)
20. Shen, G., Yang, Y., Deng, Z.: Inter-weighted alignment network for sentence pair modeling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9–11 September 2017, pp. 1179–1189 (2017)
21. Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., Cheng, X.: A deep architecture for semantic matching with multiple positional sentence representations. In: National Conference on Artificial Intelligence (AAAI), pp. 2835–2841 (2016)
22. Wang, P., Yang, Z., Niu, S., Zhang, Y., Zhang, L., Niu, S.: Modeling dynamic pairwise attention for crime classification over legal articles. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 485–494 (2018)
23. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. arXiv:Learning (2019)
24. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In: National Conference on Artificial Intelligence (2019)

25. Zhang, M., Zhou, Z.: Multilabel neural networks with applications to functional genomics and text categorization. IEEE Trans. Knowl. Data Eng. **18**(10), 1338–1351 (2006)
26. Zhong, H., Zhipeng, G., Tu, C., Xiao, C., Liu, Z., Sun, M.: Legal judgment prediction via topological learning. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3540–3549 (2018)