Hierarchical Matching Network for Crime Classification

Pengfei Wang
wangpengfei@bupt.edu.cn
School of Computer Science
Beijing University of Posts and Telecommunications

Yu Fan
adajiayou1@bupt.edu.cn
School of Computer Science
Beijing University of Posts and Telecommunications

Shuzi Niu
shuzi@iscas.ac.cn
Institute of Software
Chinese Academy of Sciences

Ze Yang
yzdestiny@gmail.com
School of Computer Science
Beijing University of Posts and Telecommunications

Yongfeng Zhang
yongfeng.zhang@rutgers.edu
Department of Computer Science
Rutgers University

Jiafeng Guo
guojiafeng@ict.ac.cn
Institute of Computing Technology
Chinese Academy of Sciences

ABSTRACT

Automatic crime classification is a fundamental task in the legal field. Given the fact descriptions, judges first determine the relevant violated laws, and then the articles. As laws and articles are grouped into a tree-shaped hierarchy (i.e., laws as parent labels, articles as children labels), this task can be naturally formalized as a two layers' hierarchical multi-label classification problem. Generally, the label semantics (i.e., definition of articles) and the hierarchical structure are two informative properties for judges to make a correct decision. However, most previous methods usually ignore the label structure and feed all labels into a flat classification framework, or neglect the label semantics and only utilize fact descriptions for crime classification, thus the performance may be limited. In this paper, we formalize crime classification problem into a matching task to address these issues. We name our model as Hierarchical Matching Network (HMN for short). Based on the tree hierarchy, HMN explicitly decomposes the semantics of children labels into the residual and alignment components. The residual components keep the unique characteristics of each individual children label, while the alignment components capture the common semantics among sibling children labels, which are further aggregated as the representation of their parent label. Finally, given a fact description, a co-attention metric is applied to effectively match the relevant laws and articles. Experiments on two real-world judicial datasets demonstrate that our model can significantly outperform the state-of-the-art methods.

CCS CONCEPTS

• **Information systems → Data mining**; • **Computing methodologies → Machine learning**; • **Applied computing → Law, social and behavioral sciences**.

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07.
https://doi.org/10.1145/3331184.3331223

KEYWORDS

Hierarchical multi-label classification, Crime Classification ,Hierarchical Matching Network

ACM Reference Format:
Pengfei Wang, Yu Fan, Shuzi Niu, Ze Yang, Yongfeng Zhang, and Jiafeng Guo. 2019. Hierarchical Matching Network for Crime Classification. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331184.3331223

1 INTRODUCTION

Crime classification is a crucial and fundamental task in the judicial field. Given the fact descriptions, one attempts to atomically determine the correct laws and articles violated, which can provide a handy reference for legal experts (e.g., lawyers and judges) and improve their working efficiency [41]. Especially crime cases are increasing in recent years, making crime classification task becomes a promising application [22].

Generally, given the fact descriptions (a set of words describing the criminal acts), judges first determine the reverent violated laws, then the articles followed. As laws and articles are grouped into a two-layers' tree hierarchy (i.e., laws as parent labels, articles as children labels) in judicial field, this task can be cast into a two-layers' hierarchical multi-label classification problem. Fig 1 gives the tree hierarchical structure over articles and laws. This specific tree-shaped structure indicates dependencies between laws and articles. Rationally utilizing these dependencies can make the classification process efficiently and effectively. For example, there are totally 452 different articles belonging to 10 laws in Chinese Criminal Law[1]. Given the fact description, judges do not need to browse all articles to make a judge for a crime, based on the tree-shaped hierarchical structure, they usually select the violated laws first, and then the relevant violated articles belonging to these laws. In addition, definitions of articles offer abundant semantics, considering these semantics can help judges make a decision accurately. This also corresponds to judges' way of working: Given the fact description, judges usually scan articles to choose the most relevant ones according to their semantics.

[1] http://www.spp.gov.cn/spp/fl/201802/t20180206_364975.shtml

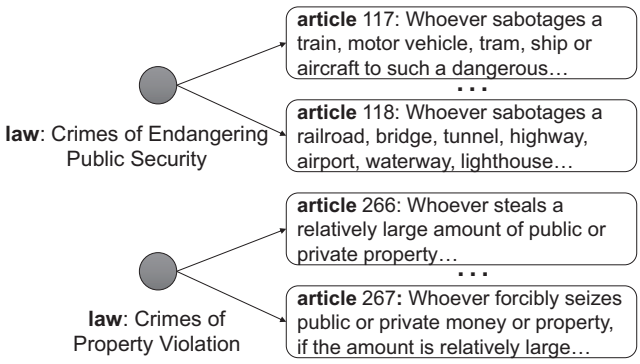

Figure 1: Hierarchical structure over laws and articles, which are extracted from Chinese Criminal Laws. Laws and articles are grouped into a tree-shaped structure, where laws are parent labels, articles are children labels. Each article only has one parent label.

Though several studies have investigated that are capable of dealing with hierarchies [7, 38], previous works in this specific scenario usually ignore the hierarchical structure over laws and articles, and feed all articles into a flat classification framework for prediction. In addition, article definitions are ignored, while these definitions offer informative semantics for classification. These two problems limit the prediction performance, which in turn raises an interesting question: Can we fuse these informative properties into a unified framework for crime classification?

To address these issues, in this paper, we design a novel Hierarchical Matching Network (HMN for short) of fusing both hierarchical structure and semantics of labels to predict correct laws and articles, and we formalize crime classification problem into a matching task between facts and labels (laws and articles). Specifically, HMN utilizes GRU to embed both labels and facts into a low embedding space. Based on the tree-shaped hierarchy over labels, HMN explicitly decomposes semantics of children labels into the alignment component and residual components according to the attention mechanism. The alignment components capture the similar semantics of children labels belonging to a same parent label, which are further aggregated as the representation of their parent label. The residual components represent the unique characteristics of each children label that are aggregated as representation of children label. Finally, a coattention mechanism is utilized between the fact labels to generate effective semantics for a correct matching.

We evaluate the effectiveness of the proposed model based on two legal datasets. For comparison, we take into account several well-known traditional flat classification models and hierarchical classification models. The empirical results show that our model can significantly outperform all the baselines in terms of all the evaluation metrics. We also provide detailed analysis on HMN model, and conduct case studies to verify the effectiveness of the decomposition strategy. In total the contributions of our work are as follows:

- We formalize crime classification problem into a matching task to analyze the semantic matching between labels (laws and articles) and facts.
- Based on the tree-shape structure and semantics of labels, we design a decomposition strategy to explicitly extract article definitions into the alignment components and residual components, both of which are further applied to generate law and article representations respectively.
- Empirically we show that our model can significantly outperform state-of-the-art baselines under different evaluation metrics on crime classification task.

The rest of the paper is organized as follows. After a summary of related work in Section 2, we give the motivation of our work in Section 3. In Section 4 we describe the problem formalization of semantic matching in juridical scenario and our proposed model. We provide experiments and evaluations in Section 5. Section 6 concludes this paper and discusses future directions.

2 RELATED WORK

In this section we briefly review three research areas related to our work, which are judgment prediction, multi-label classification, and semantic matching respectively.

2.1 Judgment Prediction

As a typical task in legal intelligence, automatic judgment prediction has been studied for decades, and most existing works usually formalize this task as the text classification framework. For example, Hu et al. [4] introduced several discriminative attributes to enhance the connections between the fact descriptions and charges, and these attributes and charges were inferred simultaneously. Inspired by the success of attention mechanism in NLP task, researchers began to handle legal prediction task by incorporating attention mechanisms. For example, Luo et al. [24] proposed an attention-based neural model for charge prediction by incorporating the relevant articles. Long el al. [23] utilized the attention mechanism to model the complex semantic interactions among facts, pleas, and articles. Wang et al. [34] introduced unified Dynamic Pairwise Attention Model for crime classification over articles. In their work, a pairwise attention model based on article definitions was incorporated into the classification model to help alleviate the label imbalance problem.

As we can see, all of these works ignored the natural tree hierarchical structure over laws and articles. Recently, Zhong et al. [27] formalized the dependencies among subtasks as a Directed Acyclic Graph and proposed a topological multi-task learning framework for both article and charge predictions. However, the valuable semantics of articles are ignored. In our work we try to fuse both hierarchical structure and semantics of labels into a unified model for crime classification.

2.2 Multi-label Classification

Multi-label classification studies the problem where each example is represented by a single instance while associated with a set of labels simultaneously. Considering the structure of labels, it can be divided into Flat Multi-label Classification, and Hierarchical Multi-label Classification [32].

2.2.1 Flat Multi-label Classification. The flat classification approach, which is the simplest one to deal with hierarchical classification

problems, consists of completely ignoring the class hierarchy, typically predicting only classes at the leaf nodes [14, 40]. For example, Boutell et al. decomposed the multi-label problem to a number of multiple dependent binary classification problems [5]. Li and Guo proposed to exploit kernel canonical correlation analysis (KCCA) to capture nonlinear label correlations and performed nonlinear label space reduction for multi-label learning [20].

2.2.2 Hierarchical Multi-label Classification. Hierarchical multi-label classification is a classification task where the classes to be predicted are hierarchically organized. Several studies have investigated new alternatives to solve HMC problems, which can be further categorized as local and global approaches [7, 31].

The idea of local approaches is to generate a hierarchy of classifiers following a top-down strategy, in which each classifier is responsible for the prediction of either particular labels or particular hierarchical levels [8, 12]. For example, Bianchi et al. [9] trained a classifier for each hierarchical label and calculated class probabilities for all examples. Wehrmann [35] proposed novel deep neural network architectures for hierarchical multi-label classification in tree-structured and DAG-structured hierarchies. Though the local approaches can well extract information from regions of the class hierarchy, a disadvantage of the local approach is that an error at a certain class level is going to be propagated downwards the hierarchy.

Global approaches usually consist of a single classifier capable of associating objects with their corresponding classes in the hierarchy as a whole. For example, Vens et al. [33] induced a single decision tree to deal with the entire class hierarchy. Chietgat [29] further used an ensemble technique to combine different decision-trees. Sangsuriyun [28] proposed a global method based on rule sets, and applied it in the classification of protein and Gene Ontology data. Compared with the local approaches, the main drawbacks of global approaches are that dependencies between classes are not leveraged in the training and classification process, and the additional computational cost of training parallel classifiers [2].

2.3 Semantic Matching

Semantic matching is a technique to identify information which is semantically related, which is widely used in question answering [21], natural language inference [6], and information retrieval [17], etc. For example, Jin et al. [18] considered a document title as a possible query, and used the title document pairs to train the translation model. Shen et al. [30] utilized the word level similarity matrix to discover fine-grained alignment of two sentences. Guo introduced a novel retrieval model by viewing the matching between queries and documents as a transportation problem [15].

The advantage of this technique is that it conducts more analysis to represent the meanings of the sentence with richer representations and then perform matching with these representations. In our work, we treat labels and facts as sentences, by this we formalize the traditional crime classification task into a matching task.

3 MOTIVATION

Article definitions contain valuable information that can help judges make a correct decision. As Table 1 shows, article 119 is determined as the semantic of its definition matches the fact description. Thus,

Table 1: An example of the judgment case, including a fact and labels violated. Words written in bold share similar semantics between the fact and labels.

fact	On the morning of May 1, 2018, local governments began to demolish illegal factories reported by the villages; by noon Guo and Wang threw several self-made **explosives** to prevent the demolition, several **cars** were **damaged**..
laws and articles	law 6: Crimes against Public Safety: article 119: Whoever **sabotages** any means of transport, **transportation facility**,electric power facility...

modelling both the label and fact semantics through a matching model seems an appropriate approach to improve the prediction performance. However, when analyzing semantics of labels, we find three interesting properties over label definitions: (1) Comparing with articles, laws have no descriptions, it brings challenges in generating semantics of laws; (2) Semantics of articles having a same parent label are similar. This is also very natural because all articles belonging to **Crimes of Endangering Public Security** are related with violence, and articles belonging to law **Crimes of Property Violation** are relevant with properties. These similar semantics over children labels can well represent their parent label to discern them from the ones belonging to other parent labels; (3) Though these similar semantics shared by its articles are useful for classifying parent labels, they become irrelevant noises for predicting the correct children labels from its siblings. For example, In Fig 1, comparing with article 266, article 267 also relates with bribery, the existence of violence is the only key factor to distinguish these two articles. These similar descriptions bring difficulties to make a correct judgment.

Thus, it is necessary to formalize crime classification task into a matching task to check whether the facts and labels are relevant according to their semantics, and a decomposition strategy is needed to extract these similar semantics over articles that shared by their sibling labels, and utilizes these similar semantics to represent their parent labels. This is the major motivation of our work.

4 OUR APPROACH

In this section, we first introduce the problem formalization of crime classification. We then describe our HMN model in detail.We finally present the learning and prediction procedure of HMN.

4.1 Formalization

Let X=$\{x_1, x_2, ...x_{|X|}\}$ denote all the facts, P=$\{p_1, p_2, \ldots, p_{|P|}\}$ denote all the parent labels (i.e, laws), and C=$\{c_1, c_2, \ldots, c_{|C|}\}$ denote all the children labels (i.e, articles). We use $C(p)$ to represent p's children labels, and $P(c)$ to represent c's parent label. Each instance is represented as a triple $(x, \mathcal{P}_x, C_x)$, where $x \in X$ represents the fact, $\mathcal{P}_x \in P$ represents the parent label set of x, and $C_x \in C$ represents the children label set of x. In the following sections, we will use the "parent label" instead of the law, and the "children label" instead of the article for clarity.

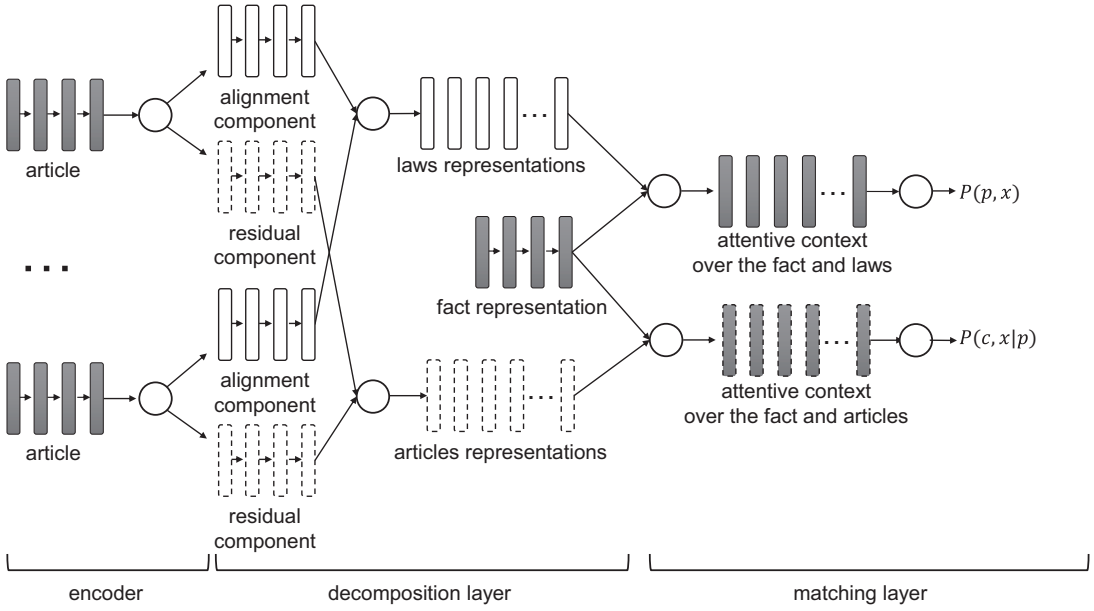

Figure 2: Over architecture of Hierarchical Matching Network (HMN). HMN contains three layers for embedding and matching. The encoder layer obtains semantic matrices of articles and facts. In decomposition layer, alignment components of articles belonging to the same law are aggregated as the law representation, while residual components are aggregated as the article representation. The matching layer generates attentive contexts, and outputs relevance scores.

Given a fact x, we aim to generate a relevance score for each parent label $p \in P$ and children label $c \in C$ to check whether they are relevant or not.

4.2 HMN

In this section, we introduce our Hierarchical Matching Network (HMN) in detail. Fig 2 shows the architecture of HMN model. The model consists of three consecutive layers for embedding and matching: 1) the encoder layer that constructs semantic matrices for both facts and children labels; 2) the decomposition layer that extracts semantic matrices of children labels into alignment components and residual components, and further generate representations of parent labels and children labels respectively; 3) the matching layer that select attentive semantics for labels and facts, and generates the relevance scores. In the following, we present the design of each layer and the philosophy of such designs.

4.2.1 ***Encoder Layer.*** In juridical field, each children label $c \in C$ and fact $x \in X$ are described by a set of words. Here we take the bag-of-word representation as the input, and map each word to a vector in a continuous space. More formally, let $\mathbf{V} = \{\mathbf{v}_i \in R^D | i = 1, 2, \dots\}$ denote all the word vectors 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 $\mathbf{h}_f(t)$ are regarded as the representation at time step t, which are obtained by the Gated Recurrent Unit [10]:

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

where n is the fixed length of $\mathbf{V}_f$. Similarly, based on GRU, given children label c' definition, we obtain the initial semantic matrix $\mathbf{S}_c$ as $[\mathbf{h}_c(1), \dots, \mathbf{h}_c(m)]$, where m is the fixed length of $\mathbf{S}_c$.

4.2.2 ***Decomposition Layer.*** Based on the initial semantic matrix of the children label generated in the previous layer, HMN decomposes each initial semantic matrix of the children label into the alignment component and the residual component explicitly. Inspired by [1, 16], for each children label c, we use the following cosine metric to measure the similarity between c and its sibling labels:

$$\mathbf{a}_{c,s} = (\mathbf{a}_{c,s}(t))_{1\times m},\ where\ \mathbf{a}_{c,s}(t) = \frac{\mathbf{h}_c(t) \cdot \mathbf{h}_s(m)}{\|\mathbf{h}_c(t)\| \cdot \|\mathbf{h}_s(m)\|} \tag{2}$$

where $s \in SIB(c)$ represents one sibling label of c. As we can see, $\mathbf{a}_{c,s}$ measures the similarity between each position of children label c and s. The alignment component of children label c is then denoted as $\mathbf{H}_{c,a} = [\mathbf{h}_{c,a}(1), \mathbf{h}_{c,a}(2), \dots, \mathbf{h}_{c,a}(m)]$ with each column $\mathbf{h}_{c,a}(t)$ computed as follows:

$$\mathbf{h}_{c,a}(t) = \frac{1}{|SIB(c)|} \sum_{s \in SIB(c)} \mathbf{a}_{c,s}(t) \cdot \mathbf{h}_c(t) \tag{3}$$

where $|SIB(c)|$ represents the number of c's sibling labels. According to Equation (3), $\mathbf{h}_{c,a}(t)$ indicates the average similarities between the t-th position of c and its sibling labels. Based on the initial semantic matrix $\mathbf{S}_c$ and the alignment component $\mathbf{H}_{c,a}$, the residual component for the children label c is defined as:

$$\mathbf{H}_{c,r} = \mathbf{S}_\mathbf{c} - \mathbf{H}_{c,a}. \tag{4}$$

As we can see, according to Equation (4), we extract the similarity semantics $\mathbf{H}_{c,a}$ from $\mathbf{S_c}$. Thus $\mathbf{H}_{c,r}$ captures the unique characteristics of label c. We then compress the alignment component $\mathbf{H}_{c,a}$ and residual component $\mathbf{H}_{c,r}$ into single vectors to summarize all the information through the aggregation function $g(\cdot)$:

$$\begin{aligned} \mathbf{v}_{c,a} = \quad g(\mathbf{H}_{c,a}) = \frac{1}{m}\sum_{t}\mathbf{h}_{c,a}(t) \\ \mathbf{v}_{c,r} = \quad g(\mathbf{H}_{c,r}) = \frac{1}{m}\sum_{t}\mathbf{h}_{c,r}(t) \end{aligned} \tag{5}$$

where $g(\cdot)$ aggregates one matrix by columns into a single vector. For a parent label p, we concatenate all the compressed alignment components and residual components of its children labels. Thus, we obtain the semantic of the parent label, and semantic of its children label set. For example, suppose the children label set of p is $C(p)$=$\{c_1, c_5, c_9\}$, the semantic matrices of p and its children set are written as $\mathbf{V}_p = [\mathbf{v}_{c_1,a}, \mathbf{v}_{c_5,a}, \mathbf{v}_{c_9,a}]$ and $\mathbf{V}_{C(p)} = [\mathbf{v}_{c_1,r}, \mathbf{v}_{c_5,r}, \mathbf{v}_{c_9,r}]$ respectively. We then iterate this procedure over all parent labels to obtain all representations of parents $\mathbf{V}_P$ and their children label sets $\mathbf{V}_C$, where $\mathbf{V}_P = [\mathbf{V}_{p_1}, \mathbf{V}_{p_2}, ..., \mathbf{V}_{p_{|P|}}]$, and $\mathbf{V}_C = [\mathbf{V}_{C(p_1)}, \mathbf{V}_{C(p_2)}, ..., \mathbf{V}_{C(C_{|P|})}]$.

As we can see, according to the decomposition layer, we obtain $\mathbf{V}_P$ to represent semantics of all laws, and $\mathbf{V}_C$ to represent semantics of all articles.

4.2.3 ***Matching Layer.*** The purpose of this layer is to select attentive semantics from both facts and labels to generate relevance scores for matching. Specifically, given a fact x, one of its parent label p, and one of its children label c $\left(c \in C_x, \text{ and } c \in C(p)\right)$, we propose a coattention mechanism [37] to compute the affinity matrix:

$$\begin{aligned} \mathbf{M}_{f,p} &= \mathbf{V}_f s(\mathbf{V}_f^T \mathbf{V}_P), \qquad \mathbf{M}_{p,f} &= \mathbf{V}_p s\big((\mathbf{V}_f^T \mathbf{V}_P)^T\big) \\ \mathbf{M}_{f,c} &= \mathbf{V}_f s(\mathbf{V}_f^T \mathbf{V}_{C(p)}), \quad \mathbf{M}_{c,f} &= \mathbf{V}_{C(p)} s\big((\mathbf{V}_f^T \mathbf{V}_{C(p)})^T\big) \end{aligned} \tag{6}$$

where s$(\cdot)$ represents the softmax function. Matrices $\mathbf{M}_{f,p}$ and $\mathbf{M}_{p,f}$ contain affinity scores between words of fact x and all parent labels P, and matrices $\mathbf{M}_{f,c}$ and $\mathbf{M}_{c,f}$ contain affinity scores between words of fact x and the children label set $C(p)$.

Note that for a children label c, as we have known its parent label through the label hierarchical structure. As a prior knowledge, we can calculate the affinity matrices according to $\mathbf{V}_{C(p)}$ instead of the whole article representations $\mathbf{V}_C$.

To integrate both the fact and label semantics for crime classification, hybrid representations $\mathbf{v}_{f,p} \in R^{2D}$ and $\mathbf{v}_{f,c} \in R^{2D}$ is then obtained from the aggregation of attention contexts. Based on the hybrid representation, our HMN outputs the relevance scores of (x, p) and (x, c) through the sigmoid function $\sigma(\cdot)$ in Equation (7).

$$\begin{aligned} P(p, x) &= \sigma(\mathbf{w}_p \cdot \mathbf{v}_{f,p}) \quad &= \sigma(\mathbf{w}_p \cdot [g(\mathbf{M}_{f,p}); g(\mathbf{M}_{p,f})]) \\ P(c, x|p) &= \sigma(\mathbf{w}_c \cdot \mathbf{v}_{f,c}) \quad &= \sigma(\mathbf{w}_c \cdot [g(\mathbf{M}_{f,c}); g(\mathbf{M}_{c,f})]) \end{aligned} \tag{7}$$

where $\mathbf{w}_p$ and $\mathbf{w}_c$ are parameters need to learn. Finally, by considering all facts and their label sets, we obtain our learning approach as follows:

$$\begin{aligned} L(\{x, \mathcal{P}_x, C_x\}) = \sum_{x \in X} \Big(\sum_{p \in \mathcal{P}_x} \big(\ln P(p, x) - \sum_{\bar{p} \in S(p)} \ln P(\bar{p}, x)\big) \\ + \sum_{c \in C_x} \big(\ln P(c, x|p) - \sum_{\bar{c} \in SIB(c)} \ln P(\bar{c}, x|p)\big)\Big) \end{aligned} \tag{8}$$

where $\bar{c}$ and $\bar{p}$ are negative labels mined from siblings of c and p respectively.

4.3 Learning and Prediction

In order to learn parameters of HMN model, we use the Adam optimizer. For each iteration, we update the parameters of our model according to Equation(8). Similar with [13, 39], we apply a simple linear function to determine a threshold for each label.

With the learned parameters, the matching strategy is as follows: For all parent labels P and children labels C, we first generate their representations through the decomposition strategy. Based on the fixed label representations, given a fact x, the best parent label set is a combination of assignments with the highest score from each parent label given the input:

$$O_p(x, p) = \sum_{p \in P} I(P(p, x) > \delta_p) \tag{9}$$

where $I(\cdot)$ denotes the indicator function, $O_p(x, p)$ is the relevance score function when feeding label set p to x, and δ_p is the learned threshold of parent label p. After obtaining the best parent label set $\mathcal{P}_x^* = \max_{p \subset P} O_p(x, p)$ according to Equation (7), the children label set is obtained as follows:

$$O_c(x, c) = \sum_{p \in \mathcal{P}_x^*} \sum_{c \in C(p)} I(P(c, x|p) > \delta_c) \tag{10}$$

where $O_c(x, c)$ is the relevance score function when feeding children labels to x, and δ_c is the learned threshold of the children label c. As we can see, according to Equation (7) and Equation (8), for each fact, we only need to conduct a forward computation to generate the scores for each parent label. Based on the selected parent labels, we further scan each of their children labels to select the relevant children labels.

5 EXPERIMENT

In this section, we conduct empirical experiments to verify the effectiveness of our proposed HMN on crime classification task. We first introduce the experimental settings, then we compare our HMN to the baseline methods to demonstrate its effectiveness on crime classification.

5.1 Dataset

We conduct our empirical experiments on two real-world legal datasets, i.e., the Fraud and Civil Action dataset, and the CAIL dataset.

- **Fraud and Civil Action** [34] comprises 40,256 criminal cases related with fraud, civil action, etc. These data are crawled from China Judgment Online[2] and span from Jan.2016 to June. 2016.
- **CAIL**[41] contains criminal cases published by the Supreme People's Court. Each case consists of two parts, i.e., fact description and corresponding judgment result (including laws, articles, and charges).

For all datasets we mentioned above, we first conduct some preprocess on our datasets. Specifically, we remove all dismissed cases. For the rest cases, we then extract fact descriptions, applicable

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

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

dataset	#fact	#Laws	#Articles	average fact description size	average article definition size	average law set size per fact	average article set size per fact
Fraud and Civil Action	17,160	8	70	1,455	136	2.6	4.3
CAIL	204,231	8	183	1,444	129	1.4	1.3

Table 3: Performance comparison over HNM-I and HMN on crime classification in terms of different evaluation metrics on two datasets. The best performance in each case is written in bold. (All the values in the table are percentage numbers with% omitted).

dataset	model	parent labels (laws)				children labels (articles)			
		Macro-P	Macro-R	Macro-F	Jaccard	Macro-P	Macro-R	Macro-F	Jaccard
Fraud and Civil Action	HMN-I	77.1	38.1	47.2	69.5	70.3	34.6	42.5	65.3
	HMN	**77.5**	**38.3**	**47.5**	**70.1**	**73.1**	**37.1**	**45.2**	**68.9**
CAIL	HMN-I	82.1	84.3	82.8	59.3	59.1	76.3	61.2	72.5
	HMN	**82.5**	**84.4**	**83.2**	**60.1**	**61.9**	**79.8**	**66.5**	**77.4**

laws and articles. After preprocessing we obtain 17, 160 facts on the Fraud and Civil Action dataset, and 204, 231 facts on the CAIL dataset. The statistics of two datasets 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.

5.2 Baselines

We adopt two types of baselines for comparison, including flat multi-label classification models and hierarchical multi-label classification models.

For flat multi-label classification models, we consider both shallow models and deep models:

- **BP-MLL**: BP-MLL [39] is derived from the popular back-propagation algorithm through employing a pairwise error function to capture the characteristics of multi-label learning.
- **CC**: Classifier Chains [26] is a chaining method that can model label correlations while maintaining an acceptable computational complexity.
- **TextCNN-MLL**: A deep flat classification method, which uses a convolution network for input representation [19], and employs a new error function similar to BP-MLL.
- **DPAM**: A unified Dynamic Pairwise Attention Model [34] that fusing article semantics into a pairwise attention matrix for crime classification.

Hierarchical multi-label classification models include:

- **HSVM**: Hierarchy of Support Vector Machine [3], where SVM is learned for each class separately, and then combined using a Bayesian network model so that the predictions are consistent with the hierarchy constraint. As we can see, HSVM is a local approach of HMC.
- **TOPJUDGE**: A topological multi-task learning framework [41], which incorporates multiple subtasks and DAG dependencies into judgment prediction.
- **HMCN**: Hierarchical Multi-label Classification Network, which is a multiple-output deep neural network that performs both local and global optimization [36].

CC, TextCNN-MLL, HSVM, and HMCN were using Scikit−multilearn[3], which is a widely adopted classification tool. For DPAM[4], BP-MLL[5], and TOPJUDGE[6], we use the code released by their authors.

5.3 Evaluation Metric

As both flat multi-label classification models and hierarchical multi-label classification models are analyzed in this paper, for fair comparison, in this paper, we employ the commonly used macro Precision (Macro-P), macro Recall (Macro-R), macro F-measure (Macro-F) and Jaccard as our evaluation metrics [5, 11, 25]. We performed significant tests using the paired t-test. Differences are considered statistically significant when the p−value is lower than 0.05.

5.4 Parameter Settings

To make fair comparisons, all the embedding parameters are randomly initialized in the range of (0, 1), the Adam optimizer is determined from 0.1 to 0.0001, and model dimension is tuned in the range of {100, 150, 200, 250, 300, 350}. For each fact description, we set n=500, for each children label definition, we set m=30.

We conduct five-fold cross-validation on the training set to tune the best hyper-parameters of each baseline. For TOPJUDGE, we use sequential form of DAG to model the dependencies between laws and articles, the learning rate of is 0.0001, and the dropout probability is 0.5. For DPAM, we set the burning number as 800, and learning rate as 0.0001. For HMN[7], we set learning rate as 0.0005. For all models, We set the batch size to 32.

[3] http://scikit.ml/
[4] https://github.com/IntelligentLaw/DPAM
[5] http://lamda.nju.edu.cn/code_BPMLL.ashx
[6] https://github.com/thunlp/TopJudge
[7] The code is available at https://github.com/IntelligentLaw/HMN

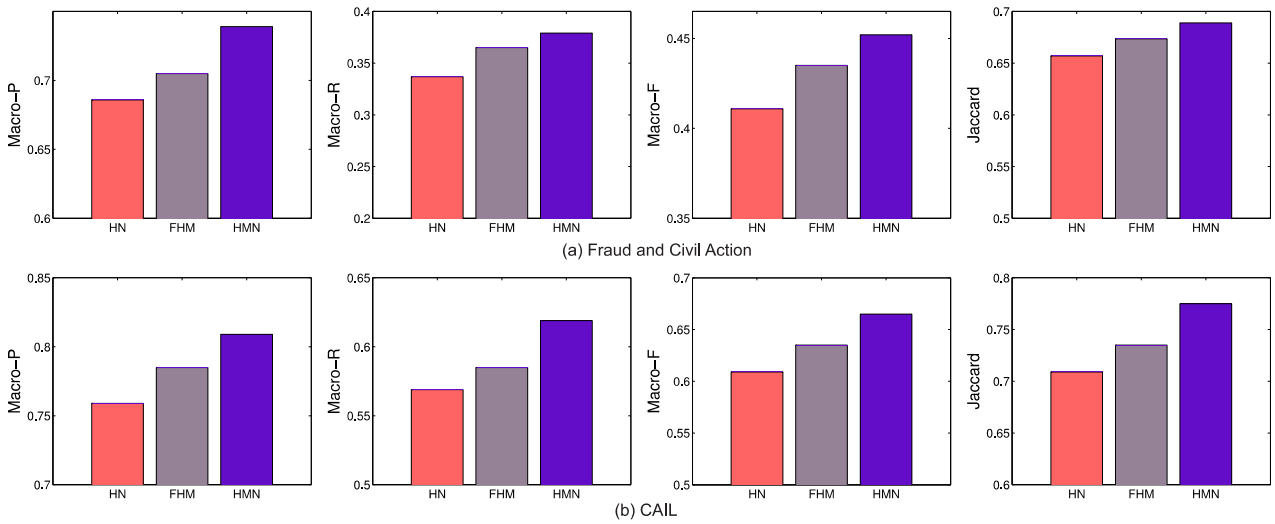

Figure 3: Performance comparison of the final HMN model with its two sub-variant models FMN and HN on two datasets in terms of Marco-P, Macro-R, Macro-F, and Jaccard.

5.5 Analysis on the HMN Model

HMN fuses both hierarchical structure and semantics of labels into a unified framework for crime classification. In this section we conducted experiments to compare different implementations of the two informative properties used in HMN. Through these experiments, we try to gain a better understanding of the model.

Table 4: Performance comparison of HNM-R and HMN over two datasets (all the values in the table are percentage numbers with% omitted). Best performance is written in bold.

dataset	model	macro-P	macro-R	macro-F	Jaccard
Fraud and Civil Action	HMN-R	69.1	34.6	42.3	66.1
	HMN	**73.1**	**37.1**	**45.2**	**68.9**
CAIL	HMN-R	76.7	53.2	60.5	73.3
	HMN	**80.9**	**61.9**	**66.5**	**77.4**

5.5.1 Analysis on Decomposition Strategy. One advantage of HMN is that it designs a decomposition strategy to generate both law and article representations. In this section we analyze the impact of decomposition strategy to crime classification.

We first make some degeneration on HMN. Specifically, in decomposition layer, for each children label $c \in C$, we replace both the alignment component $\mathbf{H}_{c,a}$ and the residual component $\mathbf{H}_{c,r}$ to $\mathbf{S}_c$. By this we use initial semantic matrices of the articles to generate laws representations and article representations respectively. We name the new model as HMN-I. Comparing with HMN, HMN-I removes the decomposition strategy, and utilizing initial label definitions to generate label representations. Table 4 shows the performance comparison between HNM and HNM-I over labels at different layers. From the results we have the following observations: (1) HMN shows a slight better performance than HMN-I on law classification over different evaluation metrics. It demonstrates that articles' alignment components have contained enough semantics to discern laws, feeding the residual components to generate law representations seems to bring none valuable information but noises for a correct law prediction. (2) HMN performs obviously better than HNM-I on article classification, the relative performance improvement on the CAIL dataset over Macro-P, Macro-R, Macro-F, and Jaccard is around 2.8%, 4.4%, 3.5%, and 4.9%, respectively. It further demonstrates that the significance of extracting similar semantics when classifying articles.

As the tree-shaped label hierarchy only has two layers, thus if children label is classified correctly. it means that all its parent labels are also classified correctly. In the following section, we only give the performance comparison over children labels.

5.5.2 Analysis on Label Definitions. Another advantage of HMN is that it introduces label definitions for crime classification. In this section we try to analyze whether introducing label definitions can bring performance for crime classification.

Specifically, we remove all article definitions from HMN, by this there is no operations for articles in both encoder layer and decomposition layer. For $\mathbf{V}_P$ and $\mathbf{V}_C$, we randomly generate each of their elements. We name the new model as HMN-R. The performance comparison between HMN and HMN-R is shown in Table 4. As we can see, HMN outperforms HMN-R on all evaluation metrics. It demonstrates the significance of introducing label definitions for crime classification.

5.5.3 Semantics VS Hierarchical Structure. In this section we further analyze the benefits when introducing label semantics and label structure respectively. For clear comparison, we also make some degradation of HMN. First, we delete the label hierarchy from HMN. Specifically, we remove laws representations from HMN, then we use the initial semantic matrices of children labels instead of residual components to generate article representations. After obtaining the hybrid representation according to the matching layer,

Table 5: Performance on crime classification between the baselines and our model (all the values in the table are percentage numbers with% omitted). The best performance in each case is written in bold. The last column shows the percentage improvement of our results against the best baseline, which are significant at p-value≤0.05.

Dataset	Metric	Flat Mulit-label Classification				Hierarchical Mulit-label Classification			Matching Model	Improve
		BP-MLL	CC	TextCNN-MLL	DPAM	HSVM	HMCN	TOPJUDGE	HMN	
Fraud and Civil Action	Macro-P	45.1	43.2	68.5	71.2	44.6	69.1	68.9	**73.1**	1.9
	Macro-R	30.4	28.6	34.3	35.5	31.5	35.3	35.1	**37.1**	1.6
	Macro-F	34.4	33.6	40.5	43.5	35.1	41.1	40.7	**45.2**	1.7
	Jaccard	60.1	58.5	65.5	67.9	62.2	66.1	65.8	**68.9**	1.0
CAIL	Macro-P	41.6	42.1	76.3	78.3	43.3	77.5	77.1	**80.9**	2.6
	Macro-R	30.2	32.5	54.3	57.7	31.2	55.6	54.9	**61.9**	3.2
	Macro-F	33.6	35.6	60.1	63.3	34.5	62.4	61.1	**66.5**	2.2
	Jaccrad	59.7	62.6	72.3	74.9	63.1	73.8	72.9	**77.4**	2.5

we use this hybrid representation to predict both of its laws and articles. We name the new model as Flat Matching Network (FMN). Secondly, in order to remove all label definitions from HMN, we directly replace hybrid representations to the fact representation in the matching layer, by this no label semantics are kept, and we name the new model as Hierarchical Network (HN). We further compare the two sub-models FMN and HN as well as our HMN to show the differences among them. Figure 3 shows the performance comparison of these three models.

As we can see, FMN performs better than HN on both datasets, it demonstrates that label semantics provide more informative properties than the hierarchical label structure for crime classification. By fusing both hierarchical structure and semantics of labels into a unified framework, HMN obtains the best performance on two datasets, which verifies the necessaries of considering both semantics and hierarchical structure of labels for crime classification.

5.6 Comparison against Baselines

We compare our HMN model to the state-of-the-art baseline methods on crime classification task. The performance results are shown in Table 5.

We first analyze the performance of flat hierarchical multi-label classification models (BP-MLL, CC, TextCNN-MLL and DPAM) on two datasets. We see that comparing with the shallow model BP-MLL and CC, the deep models TextCNN-MLL and DPAM obtain a better performance on all evaluation metrics. It demonstrates that deep models can well model the semantics of fact descriptions for classification. This observation is also coincidence with the previous findings [34, 41]. In addition, we found the performance of CC is not stable. CC works better than BP-MLL on Fraud and Civil Action dataset, while on CAIL dataset, BP-MLL can achieve a better performance than CC. The reason is that as a chaining method, CC is influenced by the error propagation: if CC misclassifies a label, the incorrect label is passed on to the next classifier and sway the next classifier to a wrong decision [34]. Finally, by fusing label semantics to alleviate the label sparsity problem, DPAM outperforms TextCNN-MLL on two datasets.

For hierarchical multi-label classification models, by leveraging label dependencies in the tree-shaped hierarchy, HSVM performs better than BP-MLL and CC. It further demonstrates the significance of concerning label hierarchical dependencies for the hierarchical multi-label classification task. By introducing label dependencies as prior knowledge for the prediction, TOPJUDGE performs better than HSVM. By simultaneously optimizing local and global loss functions for discovering local hierarchical class-relationships and global information from the entire label hierarchy, HMCN performs better than local approaches (HSVM and TOPJUDGE) on all evaluation metrics.

An interesting observation is that as a flat multi-label classification model, DPAM performs better than HMCN. The reason may be that in judicial field, comparing with label structures, label semantics play a more important role for crime classification task. This observation is also coinciding with our previous finds (e.g., FMN performs better than HN).

Finally, by fusing both hierarchical structure and semantics of labels into a unified framework, our HMN outperforms all the baseline methods in terms of all the evaluation measures on two datasets. Take the CAIL dataset as an example, when compared with the second-best baseline (i.e. DPAM), the performance improvement by HMN over Macro-P, Macro-R, Macro-F, and Jaccard is around 2.6%, 3.2%, 2.2%, and 2.5%, respectively.

5.7 The Impact of Label Definition Size

Here we investigate the impact of the label definition size to the final performance. Specifically, we tried $m \in \{10, 20, 30, 40, 50, 60, 70\}$ on two datasets. Figure 4 shows the test performance of HMN in term of Macro-F against the label definition size. From the results we find that as the label definition size m increases, the test performance in terms of Macro-F increases too. We also see that the performance begins to decrease slowly when m>30, and this trend is quite consistent on two datasets. For example, when increasing m from 50 to 60 on CAIL dataset, the relative performance decreased is about 0.021%. We assume the reason is that latter half definitions of all articles are relevant with charges (i.e., surveillance, detention, fixed-time imprisonment), which are very similar with each other. These similar definitions will be further extracted according to the decomposition layer to generate law representations, which gives no help for classifying children labels while brings more noises for law classifications. Thus, if we consider larger article definition size,

Table 6: Semantics analysis among article definitions. According to the attention weight learned in decomposition layer, words written in bold represent Top-3 significant words representing semantics of articles, while words underline are Top-3 significant words representing laws.

Law	Article	Definition
Crimes of Endangering Public Security	117	Whoever sabotages a **train**, motor vehicle, **tram**, ship or **aircraft** to such a dangerous extent as to overturn or destroy it, but with no serious consequences, shall be sentenced to fixed-term imprisonment of not less than three years but not more than 10 years...
	118	Whoever sabotages a railroad, **bridge**, tunnel, highway, **airport**, waterway, **lighthouse** or sign or conducts any other sabotaging activities to such a dangerous extent as to overturn or destroy it, but with no serious consequences, shall be sentenced to fixed-term imprisonment of not less than three years but not more than 10 years...
Crimes of Property Violation	264	Whoever **steals** a relatively large amount of public or private property or commits **theft repeatedly** shall be sentenced to fixed-term imprisonment of not more than three years, criminal detention or public surveillance and shall also, or shall only, be fined...
	267	Whoever **forcibly seizes** public or private **money** or property, if the amount is relatively large, shall be sentenced to fixed-term imprisonment of not more than three years, criminal detention or public surveillance and shall also, or shall only, be fined...

it will decrease the classification performance and bring larger computational complexity. Therefore, in our performance comparison experiment, we set article definition size as 30 on two datasets.

5.8 Case Study

To better understand what can be learned by HMN, here we conduct a case study to further analyze our decomposition strategy. Specifically, we select two articles from law **Crimes of Endangering Public Security** and two articles from law **Crimes of Property Violation** that used previously in this paper for a detail semantic analysis. As after the decomposition layer we can obtain the attention weight of each word in the article definition, here we select Top-3 words with the highest attention weights in both alignment and residual components for a detail comparison, and the result is shown in Table 6.

As we can see, for **Crimes of Endangering Public Security**, words “sabotages, overturn", and“ destroy" are chosen from its two articles (i.e., article 117, and article 118) to generate its representation, while for **Crimes of Property Violation**, words “large, private, property", and “ imprisonment" are selected. Given these words, we can easily discern these two laws. In addition, for article 264 and article 267, we see that their definitions are quite similar, the only difference is that comparing with article 264, article 267 is related with violent factors. According to the decomposition strategy, words “forcibly, seizes, money" are chosen to represent article 264, and words “steal, theft, repeatedly"" are key words representing article 267, which can also be easily discerned.

This case shows that it is necessary to extract similar semantics shared by articles having a same parent label, and our HMN can well make it through the decomposition strategy.

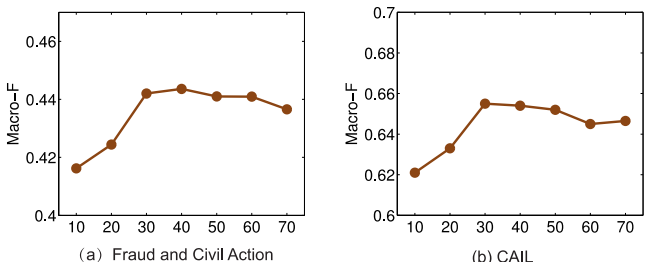

Figure 4: Performance variation in terms of Macro-F against the article definition size on two datasets. X-axis represents the definition size, which is increased from 10 to 70.

6 CONCLUSION

Crime classification is an interesting and crucial task in judicial field, which in not well explored. In this paper, we analyzed two informative properties for crime classification. In order to cast crime classification task into a matching problem for a better prediction, a novel Hierarchical Matching Network (HMN) is proposed to fuse both the hierarchical structure and semantics of labels into a unified framework. By designing a decomposition strategy, HMN decomposes article definitions into the residual and alignment components. The residual components capture unique characteristics of articles, while alignment components are aggregated to form law representations. A coattention mechanism is finally applied to generate relevance scores for matching.

In this paper, HMN concerns a shallow hierarchical structure, only dependencies between laws and articles are concerned. In the future, we will introduce more subtasks (i.e. such as charges, fines, and the term of penalty) in legal judgment, and analyze the impact of their dependencies for crime classification. We would also like to extend the usage of our HMN model to other applications to verify its effectiveness.

7 ACKNOWLEDGE

This research work was supported by the National Natural Science Foundation of China under Grant No.61802029, and the fundamental Research for the Central Universities under Grant No.500419741. We would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR* abs/1409.0473 (2014). arXiv:1409.0473 http://arxiv.org/abs/1409.0473

[2] Simon Baker and Anna Korhonen. 2017. Initializing neural networks for hierarchical multi-label text classification. In *BioNLP 2017, Vancouver, Canada, August 4, 2017*. 307–315. https://doi.org/10.18653/v1/W17-2339

[3] Zafer Barutcuoglu, Robert E Schapire, and Olga G Troyanskaya. 2006. Hierarchical multi-label prediction of gene function. *Bioinformatics* 22, 7 (2006), 830–836.

[4] Emily M. Bender, Leon Derczynski, and Pierre Isabelle (Eds.). 2018. *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*. Association for Computational Linguistics.

[5] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. 2004. Learning multi-label scene classification. *Pattern Recognition* 37, 9 (2004), 1757–1771.

[6] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 632–642. https://doi.org/10.18653/v1/D15-1075

[7] Ricardo Cerri, Rodrigo C. Barros, and André Carlos Ponce Leon Ferreira de Carvalho. 2014. Hierarchical multi-label classification using local neural networks. *J. Comput. Syst. Sci.* 80, 1 (2014), 39–56. https://doi.org/10.1016/j.jcss.2013.03.007

[8] Ricardo Cerri, Rodrigo C. Barros, and André C. P. L. F. de Carvalho. 2015. Hierarchical classification of Gene Ontology-based protein functions with neural networks. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015*. 1–8. https://doi.org/10.1109/IJCNN.2015.7280474

[9] Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. 2006. Incremental Algorithms for Hierarchical Classification. *J. Mach. Learn. Res.* 7 (Dec. 2006), 31–54. http://dl.acm.org/citation.cfm?id=1248547.1248549

[10] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *empirical methods in natural language processing* (2014), 1724–1734.

[11] Eduardo Costa, Ana Lorena, Andre Carvalho, and Alex Freitas. 2007. A review of performance evaluation measures for hierarchical classifiers. *AAAI Workshop - Technical Report* (01 2007).

[12] Eduardo P. Costa, Ana C. Lorena, André C. P. L. F. Carvalho, Alex A. Freitas, and Nicholas Holden. 2007. Comparing Several Approaches for Hierarchical Classification of Proteins with Decision Trees. In *Proceedings of the 2Nd Brazilian Conference on Advances in Bioinformatics and Computational Biology (BSB'07)*. Springer-Verlag, Berlin, Heidelberg, 126–137. http://dl.acm.org/citation.cfm?id=1776474.1776487

[13] Andre Elisseeff and Jason Weston. 2001. A kernel method for multi-labelled classification. (2001), 681–687.

[14] Eva Gibaja and Sebastian Ventura. 2014. Multilabel Learning: A Review of the State of The Art and Ongoing Research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (11 2014). https://doi.org/10.1002/widm.1139

[15] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. Semantic Matching by Non-Linear Word Transportation for Information Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*. ACM, New York, NY, USA, 701–710. https://doi.org/10.1145/2983323.2983768

[16] Hua He and Jimmy J. Lin. 2016. Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. 937–948.

[17] Posen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. (2013), 2333–2338.

[18] Rong Jin, Alex G. Hauptmann, and Cheng Xiang Zhai. 2002. Title Language Model for Information Retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02)*. ACM, New York, NY, USA, 42–48. https://doi.org/10.1145/564376.564386

[19] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *empirical methods in natural language processing* (2014), 1746–1751.

[20] Xin Li and Yuhong Guo. 2015. Multi-label classification with feature-aware non-linear label space transformation. (2015), 3635–3642.

[21] Jimmy J Lin. 2007. An exploration of the principles underlying redundancy-based factoid question answering. *ACM Transactions on Information Systems* 25, 2 (2007), 6.

[22] Shangbang Long, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2018. Automatic Judgment Prediction via Legal Reading Comprehension. *CoRR* abs/1809.06537 (2018). arXiv:1809.06537 http://arxiv.org/abs/1809.06537

[23] Shangbang Long, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2018. Automatic Judgment Prediction via Legal Reading Comprehension. *CoRR* abs/1809.06537 (2018). arXiv:1809.06537 http://arxiv.org/abs/1809.06537

[24] Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to Predict Charges for Criminal Cases with Legal Basis. *CoRR* abs/1707.09168 (2017). http://arxiv.org/abs/1707.09168

[25] Mallinali Ramírez-Corona, L. Enrique Sucar, and Eduardo F. Morales. 2016. Hierarchical Multilabel Classification Based on Path Evaluation. *Int. J. Approx. Reasoning* 68, C (Jan. 2016), 179–193. https://doi.org/10.1016/j.ijar.2015.07.008

[26] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine Learning* 85, 3 (2011), 333–359.

[27] Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). 2018. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics.

[28] Sawinee Sangsuriyun, Sanparith Marukatat, and Kitsana Waiyamai. 2010. Hierarchical Multi-label Associative Classification (HMAC) using negative rules. In *Proceedings of the 9th IEEE International Conference on Cognitive Informatics, ICCI 2010, July 7-9, 2010, Beijing, China*. 919–924. https://doi.org/10.1109/COGINF.2010.5599780

[29] Leander Schietgat, Celine Vens, Jan Struyf, Hendrik Blockeel, Dragi Kocev, and Saso Dzeroski. 2010. Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinformatics* 11 (2010), 2. https://doi.org/10.1186/1471-2105-11-2

[30] Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. 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, September 9-11, 2017*. 1179–1189.

[31] Aixin Sun and Ee-Peng Lim. 2001. Hierarchical Text Classification and Evaluation. In *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM '01)*. IEEE Computer Society, Washington, DC, USA, 521–528. http://dl.acm.org/citation.cfm?id=645496.657884

[32] Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining* 3, 3 (2007), 1–13.

[33] Celine Vens, Jan Struyf, Leander Schietgat, Saso Dzeroski, and Hendrik Blockeel. 2008. Decision trees for hierarchical multi-label classification. *Machine Learning* 73, 2 (2008), 185–214. https://doi.org/10.1007/s10994-008-5077-3

[34] Pengfei Wang, Ze Yang, Shuzi Niu, Yongfeng Zhang, Lei Zhang, and ShaoZhang Niu. 2018. Modeling Dynamic Pairwise Attention for Crime Classification over Legal Articles. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. ACM, New York, NY, USA, 485–494. https://doi.org/10.1145/3209978.3210057

[35] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical Multi-Label Classification Networks. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, StockholmsmÃdssan, Stockholm Sweden, 5075–5084.

[36] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo C. Barros. 2018. Hierarchical Multi-Label Classification Networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 5225–5234.

[37] Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic Coattention Networks For Question Answering. *international conference on learning representations* (2017).

[38] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. 1480–1489.

[39] Minling Zhang and Zhihua Zhou. 2006. Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. *IEEE Transactions on Knowledge and Data Engineering* 18, 10 (2006), 1338–1351.

[40] Minling Zhang and Zhihua Zhou. 2014. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26, 8 (2014), 1819–1837.

[41] Haoxi Zhong, Guo Zhipeng, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal Judgment Prediction via Topological Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 3540–3549. http://aclweb.org/anthology/D18-1390