Modeling Dynamic Pairwise Attention for Crime Classification over Legal Articles

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

Ze Yang
yangze01@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

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

Lei Zhang
zlei@bupt.edu.cn
School of Computer Science
Beijing University of Posts and Telecommunications

Shaozhang Niu
szniu@bupt.edu.cn
School of Computer Science
Beijing University of Posts and Telecommunications

ABSTRACT

In juridical field, judges usually need to consult several relevant cases to determine the specific articles that the evidence violated, which is a task that is time consuming and needs extensive professional knowledge. In this paper, we focus on how to save the manual efforts and make the conviction process more efficient. Specifically, we treat the evidences as documents, and articles as labels, thus the conviction process can be cast as a multi-label classification problem. However, the challenge in this specific scenario lies in two aspects. One is that the number of articles that evidences violated is dynamic, which we denote as the label dynamic problem. The other is that most articles are violated by only a few of the evidences, which we denote as the label imbalance problem. Previous methods usually learn the multi-label classification model and the label thresholds independently, and may ignore the label imbalance problem. To tackle with both challenges, we propose a unified **D**ynamic **P**airwise **A**ttention **M**odel (DPAM for short) in this paper. Specifically, DPAM adopts the multi-task learning paradigm to learn the multi-label classifier and the threshold predictor jointly, and thus DPAM can improve the generalization performance by leveraging the information learned in both of the two tasks. In addition, a pairwise attention model based on article definitions is incorporated into the classification model to help alleviate the label imbalance problem. Experimental results on two real-world datasets show that our proposed approach significantly outperforms state-of-the-art multi-label classification methods.

CCS CONCEPTS

• **Information systems → Data mining**; • **Computing methodologies → Machine learning**; • **Applied computing → Law**;

KEYWORDS

Pairwise Attention Model, Dynamic Threshold Predictor, Multi-label Classification

ACM Reference Format:
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 *SIGIR '18: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, July 8-12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209978.3210057

1 INTRODUCTION

Crimes classification over the rigorously defined legal articles is a tedious job in the juridical field. Judges usually need to consult several relevant cases to determine the specific legal articles that an evidence violated, which is time consuming and needs extensive professional knowledge. Table 1 shows an example of an evidence in a legal case, as well as the legal article that the evidence violated. Generally, the task can be cast as a multi-label classification problem to enhance working efficiency and to save manual efforts. In this work, we denote the multi-label classification problem from evidences to articles as the crimes classification task, which helps the judge to pinpoint potential articles quickly and accurately.

However, this problem is a difficult task and we may face two key challenges in practice. One is that the number of articles violated by different evidences is dynamic [10, 32, 42], i.e., the label dynamic problem. Through our analysis on a large scale real-world referee document dataset where 70 articles are considered, the article set size over evidences variants significantly, as shown in Figure 1.

The other challenge is the (class) label imbalance problem [3, 5, 34]. A multi-label classification dataset is regarded as imbalanced if some of its (minority) labels in the training set are heavily under-presented compared to other majority labels. Statistics over the same dataset is shown in Figure 2. As we can see, the number of violated evidences for each article (label) follows a long-tailed distribution, which means that many articles are seldom violated

[*] This is the corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210057

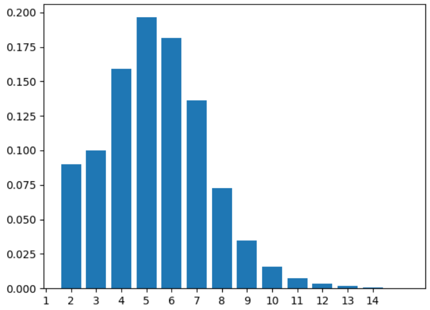

Figure 1: Distribution of article set size over evidences. x-axis stands for the article set size, y-axis indicates the proportion of evidences.

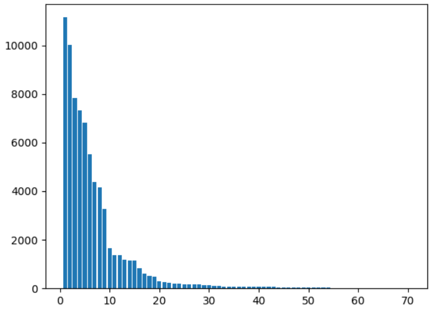

Figure 2: Distribution of article set size over evidences. The x-axis stands for the sorted labels according their frequencies in the dataset, y-axis represents counts of labels.

Table 1: An example of the judgement case, including an evidence and two articles violated.

Evidence:	*In late February 1, 2010 10 pm, Li intended to a cinema with his friend Jiang. After a contretemps with the defendant Guo, they gave Guo a beating and Guo ran away. After watched the movie, Li and Jiang were assaulted by Guo and his friends near the cinema. Jiang was stabbed by Guo.....*
Article:	***Article22**: Preparation for a crime refers to the preparation of the instruments or the creation of the conditions for a crime;* ***Article25**: A joint crime refers to an intentional crime committed by two or more persons jointly.*

by evidences. Most traditional multi-label classification algorithms try to minimize the overall classification error during the training process, which implicitly assumes equivalent importance over all labels. The skewed distribution of class labels makes classification algorithms under this equivalent assumption biased towards the majority class labels. Though article definition can indicate some relations among different articles to alleviate the label imbalance problem (as shown in Table 1, the definition of Article 22 is similar to Article 25), none of work has considered this information in crimes classification.

The difficulty in crimes classification thus raises an interesting research question: Given a set of evidences and article definitions, can we classify the evidence automatically?

Although recent studies suggest that multi-label classification is increasingly required in many applications, such as protein gene classification [2], music categorization [31], and semantic scene classification [22]. To the best of our knowledge, no practice have been conducted on crimes classification in juridical scenarios.

Previous work on multi-label classification usually exploits the label correlations, such as BP-MLL [40], kernel method [10], and calibrated label ranking [6], etc. However, all these methods learn the multi-label classification model and label threshold independently, and the label imbalance problem is largely ignored. To tackle with the first problem, we propose a multi-task framework to learn the multi-label classification model and the threshold predictor jointly. While for the second problem, we adopt the label descriptions to model the pairwise relations between labels, and extend the exact label set to a soft attention matrix over all the possible labels, which will alleviate the label imbalance problem as shown in our experiments.

In this paper we propose a unified model named **D**ynamic **P**airwise **A**ttention **M**odel (DPAM for short) for crimes classification. Specifically, we embed each evidence and article definition using the bag-of-word representations, and enumerate each article set into a pairwise label set, so that we can learn the pairwise label coverage-based classifiers from the transformed dataset. Besides, a label attention matrix is constructed based on the article definitions to alleviate the label imbalance problem. We then design a regression model to learn a multi-label threshold predictor for each label automatically. Finally, a multi-task framework is designed to learn the two tasks jointly thus to improve the generalization performance by leveraging the information contained in related tasks.

Overall, the major contributions of our work are as follows:

- We make the first attempt to investigate the prediction power of evidences and article definitions for crimes classification in juridical scenario.
- We design a multi-task learning paradigm to learn multi-label classifier and threshold predictor jointly, thus DPAM can improve the generalization performance by leveraging the information contained in related tasks.
- A Pairwise Attention Model based on article definitions is incorporated to the classification model to alleviate the label imbalance problem.
- We conduct extensive experiments on two real-world datasets to verify the effectiveness of the proposed DPAM model as compared with different baseline methods.

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

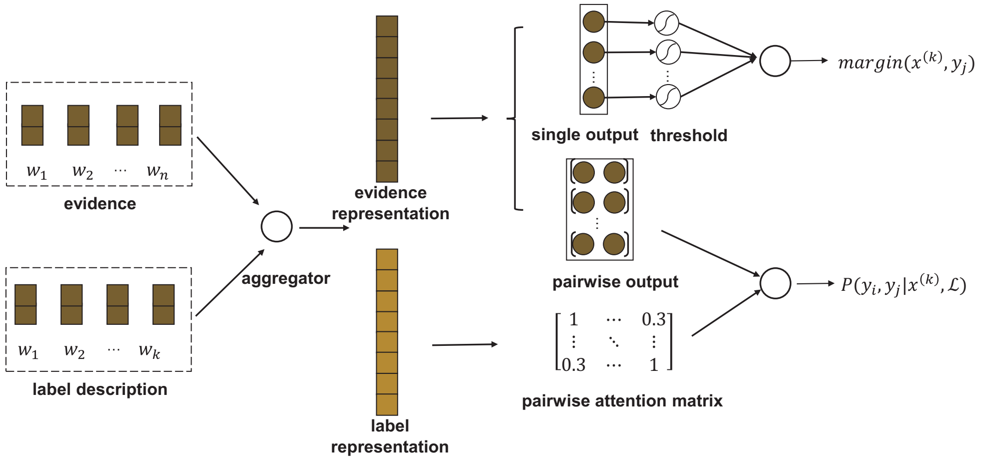

Figure 3: The overall architecture of the proposed Dynamic Pairwise Attention Model (DPAM).

2 RELATED WORK

In this section we briefly review two research areas related to our work: multi-task learning and multi-label learning.

2.1 Multi-task learning

The idea of learning multiple tasks together is to improve the generalization performance by leveraging the information contained in the related tasks. This method is widely used in various fields, such as computer vision [30, 37, 43], natural language process [8, 12, 21–23], genomics [26], demographics prediction [9, 45], and representing learning [1, 14, 44], etc. For example, Zhang et al. proposed a multi-task learning architecture with four types of recurrent neural layers to fuse information across multiple related tasks [39]. Sun et al. proposed a joint model of face identification and versification for reducing intra-personal variations while enlarging inter-personal differences [29]. Wang et al. motivated a multi-task learning-based framework for learning coupled and unbalanced representations for subspace segmentation [35]. Masaru et al. proposed a general framework of multi-task learning using curriculum learning for sentence extraction and document classification [13]. Misra et al. introduced a principled approach to learn shared representations in ConvNets using multi-task learning [25]. Pentina et al. studied a variant of multi-task learning in which annotated data is available on some of the tasks [27]. Collobert et al. introduced a single network to learn several NLP tasks jointly [8]. Liu et al. proposed an adversarial multi-task learning framework to alleviate the shared and private latent feature spaces from interfering with each other task [21]. Li et al. proposed a novel formulation by presenting a new task-oriented regularizes that can jointly prioritize tasks and instances [19]. In our model, we use a multi-task strategy to merge the evidence classifier and threshold predictor by using cross-task information.

2.2 Multi-Label Learning

Existing multi-label classification algorithms can be divided into two steps: label correlations exploitation strategies and threshold calibration learning. The first step exploits correlations among labels, and related work can be categorized into three families [42]: First-order strategy, Second-order strategy and High-order strategy. For example, Boutell et al. decomposed the multi-label problem to a number of multiple dependent binary classification problems [4]. Brinker and Klaus proposed a generic extension to overcome the expressive power limitations of previous approaches to label ranking induced by lack of calibrated scale [6]. Tsoumakas et al. proposed an ensemble method for multi-label classification [32]. In their work, a RAKEL algorithm is constructed for each member of the ensemble by considering a small random subset of labels. 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]. Zhai et al. designed an ensemble method with a minimum ranking margin objective function to construct an accurate multi-label classifier [38]. In the second step, a threshold learning mechanism is used to determine the size of label set for each instance. For example, Tsoumakas et al. used a fixed threshold to differentiate relevant and irrelevant labels for each instance [32]. Yang [36] and Fan [11] analyzed several thresholding strategies on the performance of a classier under various conditions. Elisseeff [10] and Zhang [42] designed a linear regression model to predict the label set size. As we can see, traditional methods divided the multi-label learning procedure into two independent steps (classifier learning and predictor learning), however, these two components can be very closely related in many practical tasks, thus an independent learning strategy may may limit the performance of the models.

3 OUR APPROACH

In this section, we first introduce the problem formalization of multi-label classification. We then describe the proposed DPAM model in detail. After that, we present the learning and prediction procedure of DPAM.

3.1 Formalization

We use $X = \{x^{(1)}, x^{(2)}, ..., x^{|X|}\}$ to denote all the evidences, and $C = \{y_1, y_2, ..., y_{|C|}\}$ represents the set of all possible label concepts, i.e. articles. Each $y_i \in \{0, 1\}$ indicates whether article y_i is violated or not. $|X|$ and $|C|$ represent the total number of evidences and labels. We use $\mathcal{L} = \{l^{(1)}, l^{(2)}...l^{(|C|)}\}$ to represent the label descriptions, where $l^{(i)}$ is the definition of the article y_i. For each instance $(x^{(k)}, Y^{(k)})$, we use $x^{(k)}$ to represent the k-th evidence, $Y^{(k)} \subseteq C$ represents the article set assigned to $x^{(k)}$. In the following sections, we will use "label" instead of article for clarity.

Given all the evidences X and label descriptions $\mathcal{L}$, our task is to find an optimal label set $Y^{(k)}$ for each unlabeled instance $x^{(k)}$ in the space of label sets $P(C)$, i.e. the power set of C.

3.2 DPAM

In this section, we present our Dynamic Pairwise Attention Model in detail. Figure 3 shows the architecture of our model. Specifically, our model consists of two components: a Pairwise Attention Model (PAM for short) that produces scores for labels, and a Dynamic Threshold Predictor that generates a reference point for each label to decide whether the label is relevant or not. Finally, we adopt a multi-task learning approach to learn these two tasks jointly.

3.2.1 *Pairwise Attention Model*.

In juridical field, each evidence is 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. Then we aggregate all the word vectors using some operators to form the evidence representations and label description representations. PAM considers pairwise relations between labels. Specifically, for each training instance $(x^{(k)}, Y^{(k)})$, PAM emulates all the pairwise relations in $Y^{(k)}$, by this our model can exploit the label correlations. We take $Y^{(k)} = \{y_1, y_2, y_3\}$ as an example, and after enumeration, the initial label set will be transformed into $\{(y_1, y_2), (y_1, y_3), (y_2, y_3)\}$.

More formally, let $\mathbf{V}^I = \{\vec{v}_j^I \in \mathbf{R}^{D_v} | j = 1, \ldots, N\}$ denote all the word vectors in a D_v-dimensional continuous space. For each evidence and label description, we aggregate the word vectors to form the evidence representation and label description representation separately as follows:

$$\vec{v}^{(e,k)} = g(\vec{v}_j^I : j \in x^{(k)})$$
$$\vec{v}^{(l,i)} = g(\vec{v}_j^I : j \in l^{(i)})$$

where $g(\cdot)$ denotes the aggregation function. In our work, we use TextCNN [15] to form our inputs. Given evidence $x^{(k)}$ and label descriptions $\mathcal{L}$, PAM concerns the conditional probability of pairwise $(y_i, y_j) \in Y^{(k)}$, which is written as follows:

$$P(y_i, y_j | x^{(k)}, \mathcal{L}) = P(y_i, y_j | x^{(k)}) P(l^{(i)}, l^{(j)})$$

where $P(y_i, y_j | x^{(k)})$ and $P(l^{(i)}, l^{(j)})$ is calculated separately.

To solve the label imbalance problem, we introduce the pairwise label relation. As we known, attention model in traditional sequence modeling, such as LSTM and GRU, places a soft weighting mechanism on important subsequences [33]. In order to enhance the importance of sparse pairwise labels, we extend the traditional attention model to this pairwise relation sets, namely, the **P**airwise **A**ttention **M**odel. Given label description representation $\vec{v}^{(l,i)}$ and $\vec{v}^{(l,j)}$, the pairwise attention matrix is calculated by the following function:

$$P(l^{(i)}, l^{(j)}) = \frac{\vec{v}^{(l,i)} \cdot \vec{v}^{(l,j)}}{\sum_{j=1, j\neq i}^{|C|} \vec{v}^{(l,i)} \cdot \vec{v}^{(l,j)}} \quad (1)$$

As we can see, in our model, $P(l^{(i)}, l^{(j)})$ can be regarded as an attention score to softly adjust the significance of pair (y_i, y_j) in label set $Y^{(k)}$. This mechanism will make those labels that are not in the label set also influence the final loss function, and enhance the significance of sparse pairwise labels that have similar descriptions, so that we can alleviate the label imbalance problem.

Accordingly, the posterior probability for each training label pair in the pairwise label set $P(y_i, y_j | x^{(k)})$ is calculated by a softmax function:

$$P(y_i, y_j | x^{(k)}) = \frac{exp(\vec{v}^{(e,k)} \mathbf{W} \vec{y}^{(i,j)})}{\sum_{\vec{y}^{(i,j)} \in \mathbf{Y}} exp(\vec{v}^{(e,k)} \mathbf{W} \vec{y}^{(i,j)})}$$

where $\mathbf{W} = \mathbf{R}^{D_V \times |C|}$ is the interaction matrix, $\vec{y}^{(i,j)}$ is a $|C|$ size vector, and the i-th dimension and j-th dimension of $\vec{y}^{(i,j)}$ are equal to 1, while the rest are equal to 0. $\mathbf{Y}$ is all the possible vectors when considering different pair (y_i, y_j). The objective function of PAM is then defined as the log likelihood over all the evidences as follows:

$$\begin{aligned} l_{pam} &= \sum_{x^{(k)} \in X} \sum_{(y_i, y_j) \in E(Y^{(k)})} \log P(y_i, y_j | x^{(k)}, \mathcal{L}) \\ &= \sum_{x^{(k)} \in X} \sum_{(y_i, y_j) \in E(Y^{(k)})} \Big(\log P(y_i, y_j | x^{(k)}) + \log P(l_i, l_j) \Big) \end{aligned} \quad (2)$$

where $E(Y^{(k)})$ represents the enumeration of pairwise relations in $Y^{(k)}$.

Finally, our PAM outputs the probability of each label y_i for new instance $x^{(k)}$ as the following equation based on the learned $\mathbf{W}$:

$$P(y_i | x^{(k)}) = \frac{exp(\vec{v}^{(e,k)} \mathbf{W}_{*i})}{\sum_{i=1}^{|C|} exp(\vec{v}^{(e,k)} \mathbf{W}_{*i})}$$

where $\mathbf{W}_{*i}$ represents the i-th column of $\mathbf{W}$.

3.2.2 *Dynamic Threshold Predictor*.

Through PAM, the output probability of each label $P(y_i | x^{(k)})$ is used for threshold prediction. Generally, we aim to learn a decision boundary for each label to decide whether this label is relevant to an evidence or not. Intuitively, if $P(y_i | x^{(k)})$ is above the label y_i's boundary t_i, then the label y_i is relevant to $x^{(k)}$ and $y_i \in Y^{(k)}$; if $P(y_i | x^{(k)})$ is under the y_i's boundary t_i, then the label y_i is irrelevant to $x^{(k)}$. Specifically, we use following function to measure the confidence of the predicted label for each evidence:

$$margin(x^{(k)}, y_i) = [P(y_i | x^{(k)}) - t_i)] \cdot Seg(x^{(k)}, y_i) \quad (3)$$

where $t_i \in \mathbf{T}^{1 \times |C|}$ is the boundary we need to learn for label y_i. $Seg(x^{(k)}, y_i)$ is a segmented function, which is defined as follows:

$$Seg(x^{(k)}, y_i) = \begin{cases} 1, & y_i \in Y^{(k)} \\ -1, & y_i \notin Y^{(k)} \end{cases}$$

In our model, $margin(x^{(k)}, y_i)$ represents a "safe margin" by which label y_i is relevant to evidence $x^{(k)}$. $margin(x^{(k)}, y_i) > 0$ indicates that evidence $x^{(k)}$ is correctly classified to label y_i, while

$margin(x^{(k)}, y_i) < 0$ means label y_i is irrelevant to evidence $x^{(k)}$. We the use the following function when considering each labels of all evidences:

$$l_{dyn} = \sum_{x^{(k)}\in X} \sum_{y_i \in Y^{(k)}} \log\left[1 + exp(-margin(x^{(k)}, y_i))\right] \tag{4}$$

Finally, by combining Equation(2) and Equation(4), we obtain our multi-task learning approach as follows:

$$\ell = l_{pam} + l_{dyn} - \lambda \,\|\Theta\|_2 \tag{5}$$

where λ is the regularization constant and Θ is the model parameters we need to learn (i.e. $\Theta = \{\mathbf{W}^{D_V \times |C|}, \mathbf{V}^I, \mathbf{T}^{1\times|C|}\}$).

3.3 Learning and Prediction

In order to learn parameters of DPAM model, we use the stochastic gradient decent algorithm. For each iteration, we update the parameters of our model according to Equation(5). However, the direct optimization of task PAM according Equation(2) is intractable due to the high computational cost of the normalization term which is proportional to $2^{|C|}$. Therefore, we adopt the negative sampling technique [24] for efficient optimization, which approximates the original objective l_{pam} with the following objective function:

$$\begin{aligned}\ell_{NEG} = \sum_{x^{(k)}\in X} \sum_{(y_i, y_j)\in E(Y^{(k)})} \Big(&\log\sigma(\vec{v}^{(e,k)}\mathbf{W}\vec{y}^{(i,j)}) \\ &+ n_{neg}\cdot \mathbf{E}_{\vec{y}^{neg}\sim P_{\mathcal{Y}}}[\log\sigma(-\vec{v}^{(i)\top}\mathbf{W}\vec{y}^{neg})] + \log P(l^{(i)}, l^{(j)})\Big)\end{aligned}$$

where $\sigma(x)$ is the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$, n_{neg} is the number of "negative" samples, and $\vec{y}^{neg}$ is the sampled vector, drawn according to the noise distribution $P_{\mathcal{Y}}$ which is modeled by the empirical distribution over all the possible pairwise combinations. As we can see, the objective of DPAM with negative sampling aims to differentiate the ground truth from noise by increasing the probability of the correct pairwise combination given the evidence and deceasing that of any wrong combinations. We then apply stochastic gradient descent algorithm to maximize the new objective function for learning the model.

In the training phase, we found the improvement of our model is not significant. The reason lies in the random initialization of attention matrix by aggregating the word representations. Thus, in the first a few iterations the attention matrix becomes a noise to our model. To obtain a better performance, we design a new training policy: For the first 1000 training iterations, we set $P(l^{(i)}, l^{(j)}) = 1$, and after the "burn-in" period, we assume that we have obtained the stable word representations, then we calculate our attention matrix according Equation(1) in each iteration. Details of our learning algorithm is shown in Algorithm (1):

With the learned parameters, the crimes classification strategy is as follows. For each evidence $x^{(k)}$, the best label set is a combination of assignments with the highest score from each label given the input, while satisfying that the score is larger than the label threshold. The prediction process is as follows:

$$s(Y^{(k)}|x^{(k)}) \quad = \quad \sum_{y_i \in Y^{(k)}} I\Big(\frac{exp(\vec{v}^{(e,k)}\mathbf{W}_{*i})}{\sum_{i=1}^{|C|} exp(\vec{v}^{(e,k)}\mathbf{W}_{*i})} > t_i\Big) \tag{6}$$

where $I(\cdot)$ denotes the indicator function, $s(Y^{(k)}|x^{(k)})$ is the score when feeding label set $Y^{(k)}$ to evidence $x^{(k)}$. According to Equation (6), for each evidence input, we only need to conduct a forward computation to generate the scores for each label entry, and select the combination of the highest one for each task under the condition that the score is larger than the label threshold.

Algorithm 1 Framework of joint learning for our model

```
Initialize model $\Theta = \{\mathbf{W}^{D_V\times|C|}, \mathbf{V}^I, \mathbf{T}^{1\times|C|}\}$ randomly
iter = 0
set $n_{burn}$ = 1000
repeat
    $iter \leftarrow iter + 1$
    if $iter < n_{burn}$ then
        set $P(l^{(i)}, l^{(j)}) = 1$
        for $i = 1, ..., |X|$ do
            for instance $x^{(k)}$
            compute the gradient $\nabla(\theta)$ of Equation(5)
            update model $\theta \leftarrow \theta + \epsilon\nabla(\theta)$
        end for
    else
        compute $P(l^{(i)}, l^{(j)})$ according Equation(1)
    end if
until (Coverage or $t > num$)
return $\{\mathbf{W}^{D_V\times|C|}, \mathbf{V}^I, \mathbf{T}^{1\times|C|}\}$;
```

4 EXPERIMENTS

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

4.1 Dataset

We conduct our empirical experiments over two real-world datasets from China Judgments Online[1]. China Judgments Online is a website authorized by Supreme People's Court. It records judgement documents from more than 3,000 courts across China since 2014. In this study, we collected 40256 judgement documents related with the Crime of Fraud and Civil action during the period from Jan. 2016 to June. 2016.

We first conduct some pre-process on our dataset. We remove the dismissed documents, and then we extract all the article sets and the evidences from the remaining judgement documents. After preprocessing we obtain 17,160 evidences and 70 articles on the Fraud dataset, and 4,033 evidences and 30 articles on the Civil Action dataset. The statistics of the dataset 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.

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

Table 2: Basic statistics of the two legal case datasets for experiments.

dataset	#evidence	#article	#average evidence length	#average article definition length	#average article set per evidence
Fraud	17160	70	1455	136	4.1
Civil Action	4033	30	2533	123	2.4

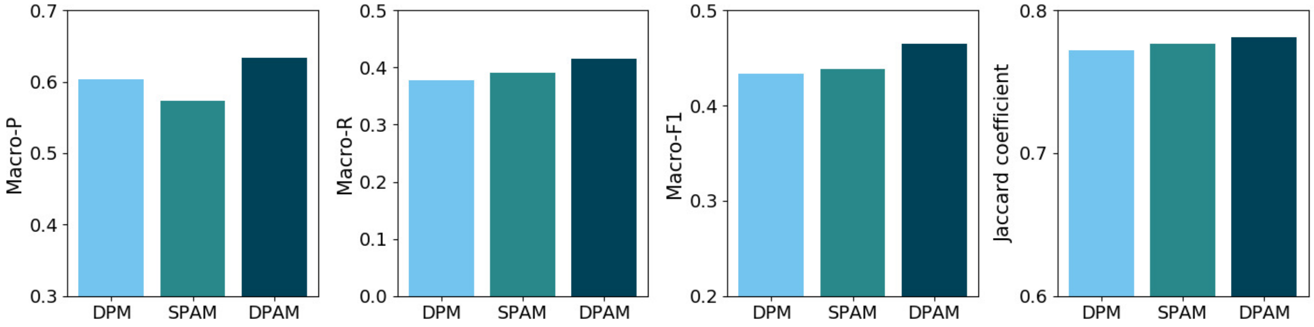

Figure 4: Performance comparison of the final DPAM model with its two sub-variant models DPM and SPAM on Fraud dataset in terms of Marco-P, Macro-R, Macro-F1, and Jaccard.

4.2 Baseline Methods

We evaluate our model[2] by comparing with several state-of-the-art methods on our dataset:

- **POP**: The top-K frequent labels in our training set is taken as the prediction for each evidence in the testing set (In our experiment, we set K=5).
- **BSVM**: A first-order multi-label method [10]. In this model, each label prediction is regarded as a binary classification problem, then a ranking approach is introduced for binary classification with SVM. For implementation, we adopt the publicly available library from *LibSVM*[3].
- **ML-KNN**: ML-KNN [41] is a popular first-order multi-label method. Based on statistical information derived from the label sets of the neighboring instances of an unseen instance, ML-KNN takes the maximum a posteriori principle to determine the label set for the unseen instance. The code is available in *sklearn*[4]
- **BP-MLL**: Backpropagation for Multi-Label Learning [40] is a popular second-order approach. It is derived from the popular Backpropagation Algorithm through employing a novel pairwise error function to capture the characteristics of multi-label learning. The code can be obtained from *lamda*[5].
- **TextCNN-MLL**: A second-order multi-label method, which uses a convolution network for input representation [15], and employs a new error function similar to BP-MLL.
- **CC**: Classifier Chains [28] is a novel chaining method that can model label correlations while maintaining an acceptable computational complexity.

For BSVM, ML-KNN, BP-MLL, TextCNN-MLL and CC, we use the publicly available PV model [18] to obtain the evidence representations. For each model, we run 20 times by setting the dimensionality $k \in \{64, 128, 192, 256, 320\}$ on both two datasets. We compare the average results of different methods and analyze the results in the following sections.

4.3 Evaluation Metrics

We use following evaluation metrics to evaluate the performance of crimes classification.

- **Jaccard similarity coefficients**: The Jaccard coefficient is a widely used multi-label classification metric [16], it measures the similarity between two label sets, and it is defined as the size of the intersection divided by the size of the union of the label sets, which is as follows:

$$Jaccard = \frac{1}{|X|}\sum_{i=1}^{|X|}\frac{|Y^{(k)} \cap Y^{(k)}_{test}|}{|Y^{(k)} \cup Y^{(k)}_{test}|}$$

  where $Y^{(k)}$ denotes the label set predicted, and $Y^{(k)}_{test}$ denotes the label set to be predicted .
- **Macro-Averaging**: The macro-average equally weights all the labels, which is computed as follows:

$$\begin{aligned} Macro\text{-}P &= \frac{1}{|C|}\sum_{j=1}^{|C|} Macro\text{-}P(j) \\ &= \frac{1}{|C|}\sum_{j=1}^{|C|}\frac{\sum_{j=1}^{|C|}|I(y_j \in Y^{(k)} \& y_j \in Y^{(k)}_{test})|}{\sum_{k=1}^{|X|}|I(y_j \in Y^{(k)})|} \end{aligned}$$

[2] https://github.com/yangze01/DPAM
[3] http://www.csie.ntu.edu.tw/čjlin/libsvm/
[4] http://lamda.nju.edu.cn/code_BPMLL.ashx
[5] http://scikit.ml/

$$\begin{aligned} Macro\text{-}R &= \frac{1}{|C|}\sum_{j=1}^{|C|} Macro\text{-}R(j) \\ &= \frac{1}{|C|}\sum_{j=1}^{|C|} \frac{\sum_{k=1}^{|X|} |I(y_j \in Y^{(k)} \& y_j \in Y_{test}^{(k)}|)}{\sum_{k=1}^{|X|} |y_j \in Y_{test}^{(k)}|} \\ Macro\text{-}F1 &= \frac{1}{|C|}\sum_{j=1}^{|C|} \frac{2 \times Macro\text{-}P(j) \times Macro\text{-}R(j)}{Macro\text{-}P(j) + Macro\text{-}R(j)} \end{aligned}$$

where I(·) is an indicator function, Macro-P(j), Macro-R(j) represent the macro precision and macro recall of the j-th label in our dataset, respectively.

4.4 Performance of two sub-models

First, we evaluate the effectiveness of the two sub-models. The purpose is to test whether it is beneficial when introducing the attention matrix and the dynamic threshold mechanism respectively. To compare we apply an uniform treatment by setting the dimensionality for both sub-models as 320, and report the results on two datasets.

4.4.1 Performance of Attention Matrix. In this section we consider the impact of attention matrix to our model. For our model DPAM, we replace the dynamic threshold mechanism by a simple Cutting Point [7, 32] procedure to determine the label set size for each evidence. We name the new model as the Static Pairwise Attention Model (SPAM for short). We further ignore the attention matrix learned by label descriptions (i.e., set a fixed score for each element in the attention matrix), and we name the degenerated model as the Static Pairwise Model (SPM for short). Table 3 shows the performance comparison of the two methods. From the results we have the following observations:(1)SPAM performs better than SPM on nearly all the evaluation metrics for both of the datasets, for example, the relative performance improvement on the Fraud dataset over Macro-R, Macro-F1, and Jaccard is around 1.8%, 1.1%, and 1.3%, respectively.(2)Comparing with SPM, the performance improvement of SPAM on Macro-P metric is slight. The underlying reason can be that though SPAM can predict more correct labels compared with SPM, it does not handle the threshold problem properly. In the prediction procedure, some unconfident labels are also recommended for each evidence, and thus the performance improvement on Macro-P is not significant.

4.4.2 Performance of Dynamic Threshold Predictor. We further analyze the impact of dynamic threshold mechanism in our model. For our model DPAM, we again make degeneration on it by ignoring the weights in the attention matrix, and the new model is denoted as the Dynamic Pairwise Model (DPM for short). We compare our DPM with several popular threshold mechanisms, i.e., the cutting point strategy and the linear mechanism, and the results are shown in Table 4. From the result we have the following observation: (1)The linear mechanism [10, 42] performs better than an ad hoc threshold calibration technique [6, 32]. (2)Our DPM performs better than linear mechanism. Take fraud dataset as an example, the relative performance improvement on Macro-P, Macro-F1, and Jaccard by our model is around 3.1%, 0.5%, and 0.5%, respectively. (3)Comparing with the linear model, we find that DPM does not achieve a significant improvement on Macro-R. The reason is that our dynamic threshold mechanism focuses on how to learn a robust threshold margin to remove the unconfident labels for each evidence, thus it tends to perform better on the Macro-P metric than the Macro-R metric.

Table 3: Performance comparison over SPM and SPAM on crimes classification in terms of different evaluation metrics. Improvements of SPAM over SPM on Macro-R, Macro-F1 and Jaccard (when applicable) are significant at $p = 0.05$.

dataset	method	Macro-P	Macro-R	Macro-F1	Jaccard
Fruad	SPM	0.572	0.372	0.430	0.768
	SPAM	0.574	0.390	0.441	0.781
Civil Action	SPM	0.645	0.322	0.424	0.623
	SPAM	0.649	0.340	0.448	0.626

Table 4: Performance comparisons over Cutting Point, linear model, and DPM on crimes classification in terms of different evaluation metrics.

dataset	method	Macro-P	Macro-R	Macro-F1	Jaccard
Fruad	Cutting Point	0.560	0.371	0.425	0.762
	Linear model	0.573	0.372	0.428	0.767
	DPM	0.604	0.377	0.433	0.772
Civil Action	Cutting Point	0.513	0.201	0.183	0.438
	Linear model	0.393	0.204	0.185	0.435
	DPM	0.653	0.329	0.457	0.613

4.5 Comparison against two sub-models

In this section, we further compare the two sub-models SPAM and DPM as well as our hybrid model DPAM to show the differences between them. Figure 4 shows the performance comparison of these three models.

An interesting observation is that SPAM obtains a better performance on Macro-R than DPM, while DPM performs better than SPAM on Macro-P. It implies that SPAM can well alleviate the label imbalance problem by introducing the attention matrix, and DPM can perform well by adjusting thresholds when predicting the label sets. Finally, by jointly learning two sub-models through a multi-task learning method, our model DPAM obtains the best performance on all evaluation metrics.

4.6 Comparison against Baselines

We further compare our model DPAM to the state-of-the-art baseline methods on crimes classification task. The performance results over the two datasets are shown in Figure 5. We have the following observations from the results:(1)It is not surprising to see that the POP method obtains the worst performance in terms of all the evaluating indicator, indicating that the crimes classification problem is not an easy task. This is due to the fact that the label set distribution in judicial field is disperse, thus predicting the same label set for each evidence is not a proper choice.(2)The first-order methods

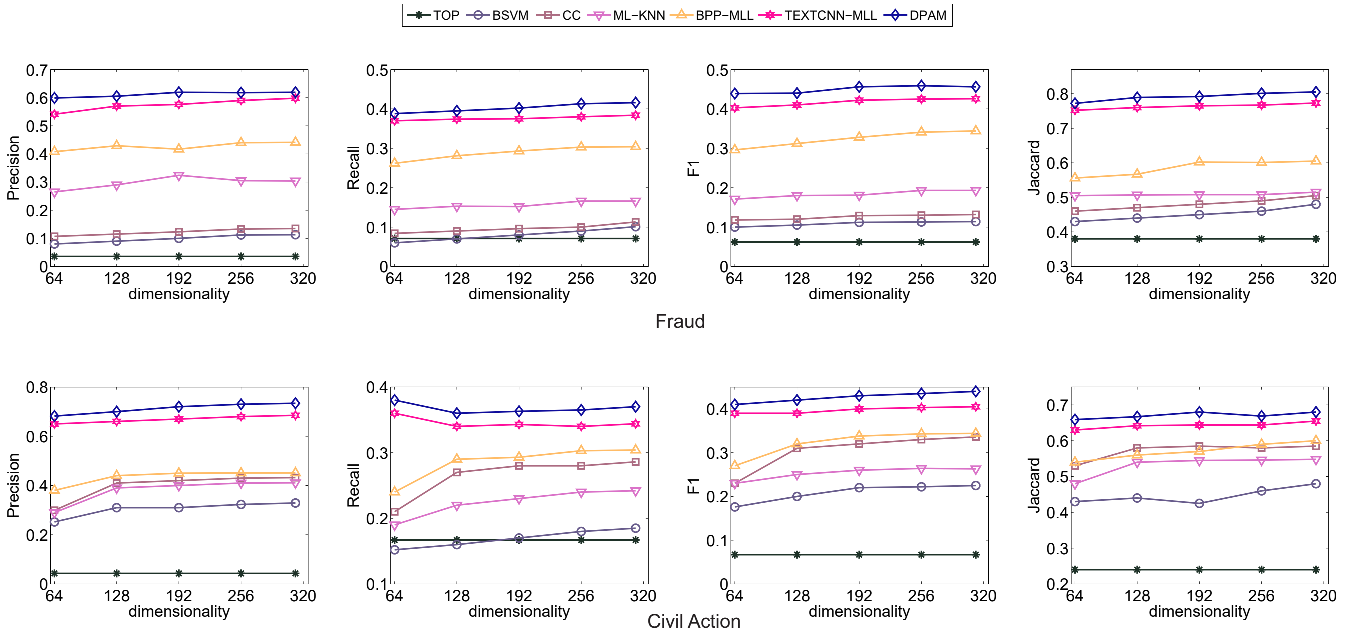

Figure 5: Performance comparison of DPAM among POP, BSVM, ML-KNN, BP-MLL, CC, and TextCNN-MLL over Fraud dataset. The dimensionality is increased from 64 to 320.

(BSVM, ML-KNN) perform better than POP method. (3)The second-order approaches perform better than the first-order approaches, and it verifies that modeling the correlation among multiple labels can improve the performance. Take Fraud dataset as an example, the relative improvement of BP-MLL over BSVM is about 24.4% in term of Macro-F1 when setting the dimensionality as 320. (4)TextCNN-MLL performs better than BP-MLL, it shows that by learning representations through a deep neural model, we can achieve a better performance than the method (BP-MLL) based on representations learned in a shallow model (i.e. PV). This result is quite consistent with the previous findings in [15].(5)CC performs better than BSVM, but with limited improvement. The reason is that as a chaining method, CC is influenced by the Error Propagation [17], i.e., when a classifier misclassifies an example, the incorrect class label is passed on to the next classifier that uses this label as an additional attribute. An incorrect value of this additional attribute may then sway the next classifier to a wrong decision.(6)Finally, when utilizing the multi-task learning paradigm to learn the threshold predictor and multi-label classification jointly, our DPAM obtains the best performance on all the evaluation metrics. For example, comparing with the second-best method (TextCNN-MLL) when setting the dimensionality as 320, the relative performance improvements of DPAM is around 2.5%, 4.3%, 3.5% and 2.0% in terms of Macro-P, Macro-R, Macro-F1, and Jaccard, respectively. The improvements are statistically significant (p-value < 0.01) over TextCNN-MLL.

4.6.1 The impact of training Policy. To learn the proposed DPAM, we utilize the burn-in procedure for optimization. One parameter in this procedure is the number of burn-ins we need to set, denoted as n_{burn}. Here we investigate the impact of the n_{burn} on the final performance.

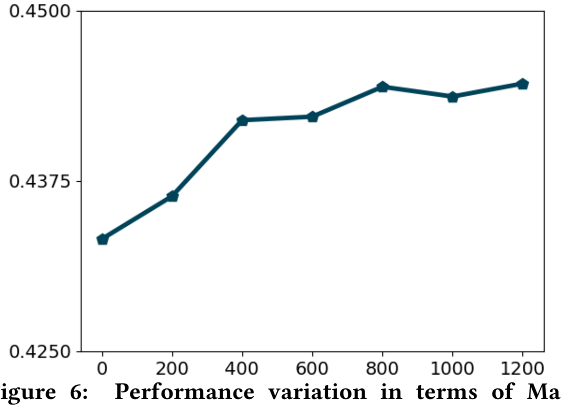

Figure 6: Performance variation in terms of Macro-F1 against the number of burn-in on two datasets. The number of burn-in is increased from 0 to 1200.

Specifically, we tried $n_{burn} \in \{0, 200, 400, 600, 800, 1000, 1200\}$ on the Fraud dataset. Figure 6 shows the test performance of DPAM in term of Macro-F1 against the number of burn-in when setting the dimensionality as 320.

From the results we find that: (1) As the burn-in number n_{burn} increases, the test performance in terms of Macro-F1 increases too.(2)As the burn-in number n_{burn} increases, the performance gain between two consecutive trials decreases. For example, when we increase n_{burn} from 800 to 1000, the relative performance improvement in terms of Macro-F1 is about 0.3%. It indicates that after 800 iterations, we have obtained stable word representations, and if

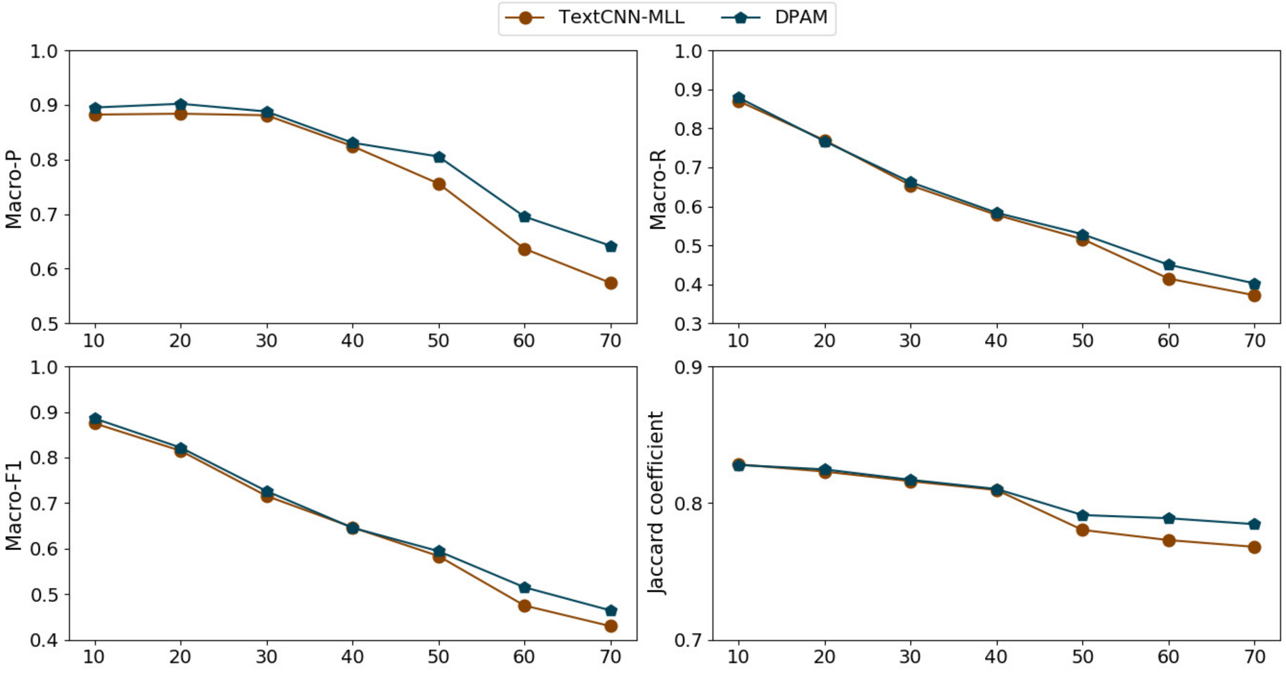

Figure 7: Performance comparison among different label group size. The x-axis represents the label size modeled, y-axis represents the performance in terms of different evaluations metrics.

we continue to burn more iterations, there will be less performance improvement but larger computational complexity. Therefore, in our performance comparison experiment, we set n_{burn} as 1000 on the Fraud dataset, and results are similar on the civil action dataset.

4.7 Case Study

To obtain a better understanding why DPAM performs better than other models, in this section, we conduct the case study to compare DPAM and the second-best model TextCNN-MLL qualitatively. Take Fraud dataset as an example, we first sort all the 70 articles according their frequency of occurrence in our dataset, then we split the sorted labels into 7 groups, where each group contains 10 labels. In this way, the first group contains the most frequent 10 labels, while the 7-th group contains the sparsest 10 labels.

Given this, we compare the two models mentioned above on the first group, and we repeat the process six times, each time we add the next label group into comparison. By this we want to test whether DPAM can perform well when faced with the label imbalance problem. The results are shown in Figure 7, and we have the following observations:(1)The performance of DPAM and TextCNN-MLL decrease when considering more labels, and this is consistent with the expectation that feeding sparse labels will degrade the performance.(2)Comparing with TextCNN-MLL, DPAM shows no significant improvement on all evaluation metrics when modeling the first 4 label groups, and this verifies that the attention matrix is not working when all of the labels occur frequently in the dataset. (3)DPAM outperforms TextCNN-MLL in all the evaluation metrics since we add the 5-th group. An interesting observation is that performance gain between DPAM and TextCNN-MLL is increasing when adding the remaining groups one by one. It implies that DPAM can alleviate the label imbalance problem by introducing the attention matrix into the modeling.

5 CONCLUSION

In this paper, we address the problem of crimes classification in juridical scenario, and we cast it as the multi-label problem. A Dynamic Pairwise Attention Model (DPAM for short) is proposed to predict the article set for each evidence. By introducing an attention matrix learned from article definitions, our model can alleviate the label imbalance problem. A dynamic threshold predictor mechanism is further proposed to learn a robust threshold for each article atomically. Finally, we adopt the multi-task learning paradigm to learn multi-label classification and the threshold predictor jointly, which can improve the generalization performance by leveraging the information contained in the two tasks. We conduct experiments on two real-world datasets, and verified that our approach can outperform many state-of-the-art baseline methods consistently under different evaluation metrics.

In DPAM, we used a TextCNN to obtain the evidence representations. However, in juridical field, some keywords in evidence, such as murder, robbery, are also valuable for judges to classify the evidences. Feeding these keywords with other words in evidences into a united model may weaken the significance of the keywords. In the future, we will analyze the significance of keywords to crimes classification, and it would be interesting to analyze the interactions between the keywords and the evidences.

ACKNOWLEDGEMENT

This research work was supported by the fundamental Research for the Central Universities, the National Natural Science Foundation of China under Grant No.61602451,the Joint Funds of NSFC-Basic Research on General Technology under Grant No.U1536121. We would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2007. Multi-task feature learning. In *Advances in neural information processing systems*. 41–48.

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

[3] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. 2004. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explor. Newsl.* 6, 1 (June 2004), 20–29. https://doi.org/10.1145/1007730.1007735

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

[5] Paula Branco, Luis Torgo, and Rita P Ribeiro. 2015. A Survey of Predictive Modelling under Imbalanced Distributions. *arXiv: Learning* (2015).

[6] Klaus Brinker. 2008. Multilabel classification via calibrated label ranking. *Machine Learning* 73, 2 (2008), 133–153.

[7] Amanda Clare and Ross D King. 2001. Knowledge Discovery in Multi-label Phenotype Data. *european conference on principles of data mining and knowledge discovery* (2001), 42–53.

[8] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, 160–167.

[9] Yuxiao Dong, Yang Yang, Jie Tang, Yang Yang, and Nitesh V Chawla. 2014. Inferring user demographics and social strategies in mobile social networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 15–24.

[10] Andr Elisseeff and Jason Weston. 2001. A kernel method for multi-labelled classification. In *International Conference on Neural Information Processing Systems: Natural and Synthetic*. 681–687.

[11] Rong-En Fan and Chih-Jen Lin. 2007. A study on threshold selection for multi-label classification. *Department of Computer Science, National Taiwan University* (2007), 1–23.

[12] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: a deep learning approach. In *International Conference on International Conference on Machine Learning*. 513–520.

[13] Masaru Isonuma, Toru Fujino, Junichiro Mori, Yutaka Matsuo, and Ichiro Sakata. 2017. Extractive Summarization Using Multi-Task Learning with Document Classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. 2091–2100.

[14] Zhuoliang Kang, Kristen Grauman, and Fei Sha. 2011. Learning with Whom to Share in Multi-task Feature Learning. In *International Conference on Machine Learning, ICML 2011, Bellevue, Washington, Usa, June 28 - July*. 521–528.

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

[16] Oluwasanmi O Koyejo, Nagarajan Natarajan, Pradeep K Ravikumar, and Inderjit S Dhillon. 2015. Consistent Multilabel Classification. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 3321–3329. http://papers.nips.cc/paper/5883-consistent-multilabel-classification.pdf

[17] Miroslav Kubat. 2017. Induction in Multi-Label Domains. (09 2017), 251-271 pages.

[18] Quoc V Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *international conference on machine learning* (2014), 1188–1196.

[19] Changsheng Li, Junchi Yan, Fan Wei, Weishan Dong, Qingshan Liu, and Hongyuan Zha. 2016. Self-Paced Multi-Task Learning. *national conference on artificial intelligence* (2016), 2175–2181.

[20] Xin Li and Yuhong Guo. 2015. Multi-label classification with feature-aware non-linear label space transformation. In *International Conference on Artificial Intelligence*. 3635–3642.

[21] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-task Learning for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. 1–10.

[22] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye Yi Wang. 2015. Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification avvnd Information Retrieval. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 912–921.

[23] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114* (2015).

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv: Computation and Language* (2013).

[25] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3994–4003.

[26] Guillaume Obozinski, Ben Taskar, and Michael I Jordan. 2010. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing* 20, 2 (2010), 231–252.

[27] Anastasia Pentina and Christoph H Lampert. 2017. Multi-Task Learning with Labeled and Unlabeled Tasks. *stat* 1050 (2017), 1.

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

[29] Yi Sun, Xiaogang Wang, and Xiaoou Tang. 2014. Deep Learning Face Representation by Joint Identification-Verification. *Advances in Neural Information Processing Systems* 27 (2014), 1988–1996.

[30] Antonio Torralba, Kevin P Murphy, and William T Freeman. 2007. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 5 (2007), 854–869.

[31] Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris, and Ioannis P. Vlahavas. 2008. Multi-label classification of music into emotions. In *Ismir 2008, International Conference on Music Information Retrieval, Drexel University, Philadelphia, Pa, Usa, September*. 325–330.

[32] Grigorios Tsoumakas and Ioannis Vlahavas. 2007. Random k-Labelsets: An Ensemble Method for Multilabel Classification. In *European Conference on Machine Learning*. 406–417.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR* abs/1706.03762 (2017). arXiv:1706.03762 http://arxiv.org/abs/1706.03762

[34] Byron C. Wallace, Kevin Small, Carla E. Brodley, and Thomas A. Trikalinos. 2011. Class Imbalance, Redux. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining (ICDM '11)*. IEEE Computer Society, Washington, DC, USA, 754–763. https://doi.org/10.1109/ICDM.2011.33

[35] Yu Wang, David Wipf, Qing Ling, Wei Chen, and Ian Wassell. 2015. Multi-task learning for subspace segmentation. In *International Conference on International Conference on Machine Learning*. 1209–1217.

[36] Yiming Yang. 2001. A Study of Thresholding Strategies for Text Categorization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '01)*. ACM, New York, NY, USA, 137–145. https://doi.org/10.1145/383952.383975

[37] Junho Yim, Heechul Jung, ByungIn Yoo, Changkyu Choi, Dusik Park, and Junmo Kim. 2015. Rotating your face using multi-task deep neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 676–684.

[38] Shaodan Zhai, Chenyang Zhao, Tian Xia, and Shaojun Wang. 2015. A Multi-label Ensemble Method Based on Minimum Ranking Margin Maximization. In *IEEE International Conference on Data Mining*. 1093–1098.

[39] Honglun Zhang, Liqiang Xiao, Yongkun Wang, and Yaohui Jin. 2017. A Generalized Recurrent Neural Architecture for Text Classification with Multi-Task Learning. (2017), 3385–3391.

[40] Min Ling Zhang and Zhi Hua 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.

[41] Min-Ling Zhang and Zhi-Hua Zhou. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition* 40, 7 (2007), 2038–2048.

[42] Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering* 26, 8 (2014), 1819–1837.

[43] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. 2013. Robust Visual Tracking via Structured Multi-Task Sparse Learning. *International Journal of Computer Vision* 101, 2 (2013), 367–383.

[44] Yu Zhang, Dityan Yeung, and Qian Xu. 2010. Probabilistic Multi-Task Feature Selection. *Advances in Neural Information Processing Systems* (2010), 2559–2567.

[45] Erheng Zhong, Ben Tan, Kaixiang Mo, and Qiang Yang. 2013. User demographics prediction based on mobile data. *Pervasive and Mobile Computing* 9, 6 (2013), 823–837.