Learning Legal Text Representations via Disentangling Elements

Yingzhi Miao[a] (yzmiao21@stu.ecnu.edu.cn), Fang Zhou[a] (fzhou@dase.ecnu.edu.cn), Martin Pavlovski[b] (martin.pavlovski@temple.edu), Weining Qian[a] (wnqian@dase.ecnu.edu.cn)

[a] East China Normal University, Shanghai, 200062, China
[b] Temple University, Philadelphia, 19122, USA

Corresponding Author:
Fang Zhou
East China Normal University, Shanghai, 200062, China
Email: fzhou@dase.ecnu.edu.cn

Abstract

Recently, a rising number of works has been focusing on tasks in the legal field for providing references to professionals in order to improve their work efficiency. Learning legal text representations, being the most common initial step, can strongly influence the performance of downstream tasks. Existing works have shown that utilizing domain knowledge, such as legal elements, in text representation learning can improve the prediction performance of downstream models. However, existing methods are typically focused on specific downstream tasks, hindering their effective generalization to other legal tasks. Moreover, these models tend to entangle various legal elements into a unified representation, overlooking the nuances among distinct legal elements. To solve the aforementioned limitation, we (1) introduce a generic model, called *eVec* (legal text to element-related Vector), based on a triplet loss to learn discriminative representations of legal texts concerning a specific element, and (2) present a framework *eVecs* for learning disentangled representations w.r.t. multiple elements. The learned representations are independent of each other in terms of elements, and can be directly applied to or fine-tuned for various downstream tasks. We conducted comprehensive experiments on two real-world legal applications, the results of which indicate that the proposed model outperforms a range of baselines by a margin of up to 34.2% on a similar case matching task and 14% on a legal element identification task. When a small quantity of labeled data is accessible, the proposed model's superior performance becomes even more evident.

Keywords:
Legal text representations, Elements, Disentangled representations

1. Introduction

Nowadays, a growing number of research has focused on legal text mining for tasks such as charge prediction (Long et al., 2019; Zhang et al., 2023; Liu et al., 2022) and similar case matching (Bhattacharya et al., 2022; Charmet et al., 2022; Peng et al., 2020), which are becoming vital parts of legal assistant systems. Such techniques could improve the efficacy of legal experts while also assist people who lack legal knowledge and are unfamiliar with complex legal procedures to obtain convenient and high-quality inquiry services.

Table 1: Crucial legal elements in Chinese private lending cases (Xiao et al., 2019).

Element	Examples
贷款人和借款人的性质 (Properties of lender and borrower)	自然人，法人等 (a natural person, a legal person, etc.)
保证类型 (Type of guarantee)	保证，不保证，质押等 (guarantee, no guarantee, pledge, etc.)
借款用途 (Usage of the loan)	事业、个人生活、家庭生活等 (personal life, family life and business, etc.)

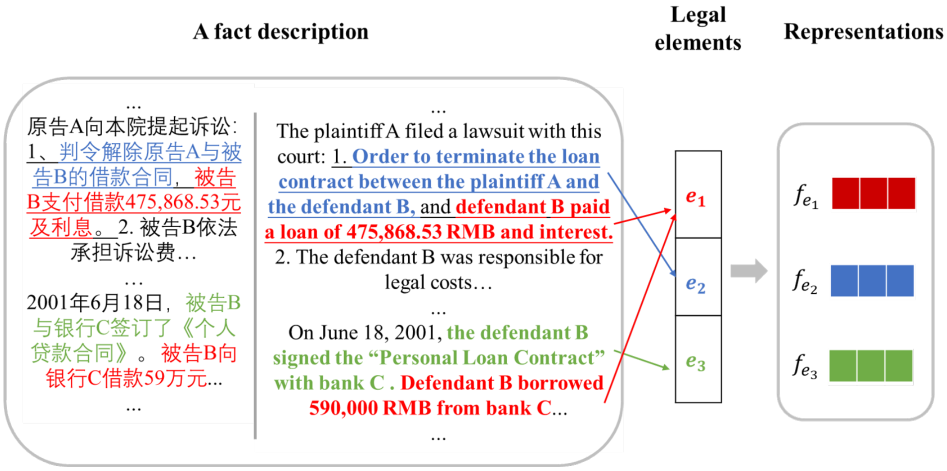
As legal assistant systems take fact descriptions of legal cases as input, the first step in general is to construct a semantic representation of a fact description and then input it into a model designed for a downstream task. The traditional text representation methods used for legal text[1] analysis focus on extracting relevant features from a text. For example, Tran et al. (2019) extracted textual features, such as word, phrase, sentence and summary, and Katz et al. (2017) selected features such as dates, location, terms and types from case profiles. However, these feature-based methods are limited to learning literal representations of legal texts, while ignoring vital legal knowledge in the legal texts.

Recently, legal knowledge, such as law articles or legal elements, has been integrated into the proposed models (Gan et al., 2021; Long et al., 2019; Zhang et al., 2023). Compared with law articles[2], legal elements can be considered as more fine-grained domain knowledge, particularly in Chinese legal texts. Legal elements are features of fact descriptions which might affect legal consequences (Peng et al., 2020; Xiao et al., 2019; Wang, 2022). Table 1 shows some crucial legal elements in Chinese private lending cases.

Preserving element information in the representation of a Chinese legal text is critical for downstream tasks in the legal domain. Note that a Chinese legal text may contain several elements that are tightly coupled. The existing representation methods (Peng et al., 2020; Wang, 2022) usually compress the information of all elements into a unified vector. However, elements may play distinct roles in different tasks. Using a single vector to represent all elements' information may weaken the effect of an individual element. For example, when considering similar Chinese private lending cases, legal experts are more concerned about the qualification of a lender such as whether s/he has a proof of loan (e_2) or is a financial institution (e_3); than the loan amount (e_1) (refer to Fig. 1). Using a single vector that simply aggregates the information of the three elements may weaken

[1] The phrase "legal text" used in this paper is equivalent to "fact description".

[2] Law articles are legal principles that are cited by legal professionals to support their arguments.



...
原告A向本院提起诉讼：1、判令解除原告A与被告B的借款合同，被告B支付借款475,868.53元及利息。2. 被告B依法承担诉讼费…
…
2001年6月18日，被告B与银行C签订了《个人贷款合同》。被告B向银行C借款59万元...
...

...
The plaintiff A filed a lawsuit with this court: 1. **Order to terminate the loan contract between the plaintiff A and the defendant B,** and **defendant B paid a loan of 475,868.53 RMB and interest.** 2. The defendant B was responsible for legal costs…
...
On June 18, 2001, **the defendant B signed the "Personal Loan Contract" with bank C . Defendant B borrowed 590,000 RMB from bank C**...
...

Figure 1: The texts highlighted in red are related to the element e_1 "借款金额n万元" (loan n ten thousand yuan. The text highlighted in blue is related to the element e_2 "有借贷证明" (have proof of loan), and the text highlighted in green is related to e_3 "贷款人系金融机构" (the lender is a financial institution). The proposed model identifies the information related to e_1, e_2, e_3 and then generates three corresponding representations, $f_{e_1}, f_{e_2}, f_{e_3}$.

the importance of e_2 and e_3. Hence, the first challenge is *how to better represent the information about an individual element.*

The second challenge is *how to capture nuances among elements.* Some elements are composed of similar Chinese characters but have distinct meanings. For instance, two elements, "经济补偿金" (economic compensation) and "经济赔偿金" (economic damage compensation) are almost the same except for one character, but have totally different meanings. Since both elements, for the most part, consist of identical Chinese characters, traditional content-based text representation methods (Blei et al., 2003) will neutralize the subtle difference and produce similar representations.

Moreover, when dealing with a relatively small labeled dataset, the model may be susceptible to overfitting, leading to poor generalization capabilities. However, annotating legal elements requires annotators to possess a certain level of legal domain knowledge, which can contribute to increased labor costs associated with labeling. Hence, the third challenge is *how to enhance the model's generalization ability in the presence of a limited amount of labeled data.*

Specifically, the work's **research goal** is to learn disentangled representations of fact descriptions w.r.t. different legal elements to be able to (1) maximally preserve the information relevant to a specific element, (2) capture the subtle differences between different elements, and (3) achieve effective generalization even in cases with a limited amount of annotated data.

To mitigate the aforementioned three challenges, we propose a task-agnostic supervised neural model, named as *eVec* (legal text to element-related Vector), based on a triplet loss for learning discriminative embeddings of fact descriptions w.r.t. a specific element. Given an element e and a fact description x that contains information about e, the goal is to learn a representation of x which can preserve more semantic information w.r.t. e. **The proposed model can learn a disentangled representation of a fact description concerning a specific element.** Furthermore, owing to its task-agnostic nature, **the proposed model can be utilized for diverse legal tasks across various domains of law**.

For multi-element tasks (see Fig. 1), we design a framework *eVecs*, which ensembles multiple *eVec* models, to learn representations of a fact description w.r.t. each element independently. The learned representations can be directly applied to or fine-tuned for downstream tasks. Unlike the existing representation learning methods (Hu et al., 2018; Kim, 2014; Wang, 2022) which entangle the information of all elements into one vector, the *eVecs* framework is capable of learning representations that not only preserve all elements' information but also well separate them. **The learned representations are independent of each other in terms of elements.** Additionally, *eVecs* demonstrates substaintial improvements when dealing with a limited amount of labeled data per element.

The contributions are summarized as follows:

- To our knowledge, the concept of learning disentangled representations of fact descriptions has not been explored. We introduce a supervised neural model *eVec* based on a triplet loss for learning representations of fact descriptions. It aims to learn a disentangled representation that **maximally preserves the information relevant to a specific element while omitting irrelevant element information.**

- The learned representations are **task-independent** and can be directly applied to multiple tasks or fine-tuned for different downstream tasks.

- The results on two real-world applications, similar case matching and legal element identification, show that our model outperforms other baselines. The benefit of our model is even more noticeable when **only a small quantity of labeled data is accessible**.

2. Related work

2.1. Text representation

In the analysis of legal documents, earlier works mainly focused on extracting textual features of texts (Katz et al., 2017; Liu & Hsieh, 2006; Sulea et al., 2017; Tran et al., 2019; Yan et al., 2017). However, these methods only capture shallow textual features. Designing and annotating features require massive human efforts. Considering the limitations of feature-based methods, recent studies incorporate legal knowledge into classical deep neural networks. For example, Long et al. (2019) utilizes legal knowledge as an auxiliary input. Other studies introduce multi-task frameworks to model the target task and the domain-dependent task together, such as relevant law article extraction (Feng et al., 2022; Ma et al., 2021; Yang et al., 2019; Zhang et al., 2023).

A few recent works have taken legal elements into account. For example, Peng et al. (2020) presents a text matching model based on element extraction. Each dimension of the element representation is the likelihood that a sentence contains an element. Hu et al. (2018) adopts an attention mechanism to learn an element representation, which is obtained by summing weighted word embeddings. Note that the concept of elements in Zhong et al. (2020) is different from ours, as the elements they considered are the subject, the object and the motive of a law article.

Some other studies have proposed approaches for the task of legal case retrieval (Shao et al., 2020; Chalkidis et al., 2020; Vuong et al., 2022). However, it is worth noting that most of the cases they investigated stem from common law systems, where judgments rely on the outcomes of prior case decisions. In contrast, the legal cases in this study are derived from civil law systems, where judgments are based on codified legal provisions. As a result, some of these approaches may not be directly applicable to the datasets used in this study.

Despite the aforementioned advances, the existing works fail to capture the subtle differences between elements. To deal with these issues, we introduce a novel model to learn disentangled representations of legal texts w.r.t. elements, where representations of legal texts containing the same element are closer together but farther apart from texts not containing it.

2.2. Disentangled representation learning

There has been an increasing interest in disentangled representation learning in recent years. The objective of disentangled representation learning is to distinguish the attributes of input data and map them into different independent latent spaces.

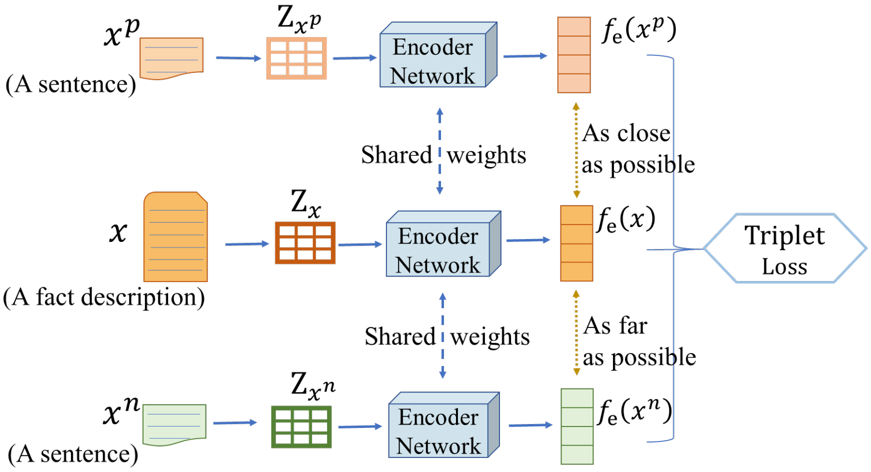


Figure 2: An illustration of the proposed model *eVec*.

The idea of disentangled representations has been explored first in computer vision (Chen et al., 2016; Denton & Birodkar, 2017; Mathieu et al., 2016). Recently, some researchers focused on studying disentangled representations in NLP tasks, such as text style transfer (Cheng et al., 2020; John et al., 2019; Yang et al., 2018; Zhu et al., 2022), semantic parsing (Yin et al., 2018) and text generation (Fei et al., 2022; Wiseman et al., 2018). The most commonly used approach among them is based on adversarial training. Inspired by the work in Jain et al. (2018), which proposes to applied adversarial learning on the (dis)similarity triplets of biomedical abstracts w.r.t. specific aspects, our work extends this approach to the legal domain to learn element-specific representations of legal texts. The learned representations are task-independent and can be directly applied to downstream element-related tasks.

3. Method

In this section, we first provide a description of the proposed model *eVec* that maps a fact description to a low-dimensional embedding space w.r.t. one element, and then present a general framework, *eVecs*, which learns disentangled representations of a legal text w.r.t. multiple elements.

3.1. eVec: legal text to element-related vector

The fact description of a legal case describes the history of the dispute and is composed of several sentences. Assume a fact description $x = \{w_1, ..., w_N\}$ and an element e, where w_t represents the t-th word in x and N is the number of words in x. The goal is to learn a discriminative embedding $f_e(x)$ of x w.r.t. e. We utilize additional texts to help the model better analyze the information that is relevant to the element e in x. Namely, let x^p represent a sentence containing the element e, labeled with a positive class, and x^n represent a sentence that does not contain any description relevant to e, labeled with a negative class. The hypothesis is that the intra-class distance between $f_e(x)$ and $f_e(x^p)$ is smaller compared to the inter-class distance between $f_e(x)$ and $f_e(x^n)$.

3.1.1. Triplet loss

Let ν be the collection of all feasible triplets in the training dataset and let D denote its cardinality. For the i-th triplet (x_i, x_i^p, x_i^n), the goal of the *eVec* model is to ensure that the embedding of x_i is closer to the embedding of x_i^p than to the embedding of x_i^n, that is,

$$sim(f_e(x_i), f_e(x_i^p)) - \alpha > sim(f_e(x_i), f_e(x_i^n)), \forall (f_e(x_i), f_e(x_i^p), f_e(x_i^n)) \in \nu, \tag{1}$$

where α is a margin that is applied to make a certain distance between the positive and negative pairs and $sim(f_e(x_i), f_e(x_i^p))$ is the cosine similarity between $f_e(x_i)$ and $f_e(x_i^p)$.

The ultimate objective is to minimize the loss $\mathcal{L}$,

$$\mathcal{L} = \sum_{i=1}^{D} [max(0, \alpha - sim(f_e(x_i), f_e(x_i^p)) + sim(f_e(x_i), f_e(x_i^n)))]. \tag{2}$$

Due to the margin threshold α, the triplet loss will enforce large distances between the positive and negative pairs. Other studies (D'Innocente et al., 2021; Wang et al., 2014) have shown that using triplet loss helps capture nuances among inputs compared to using contrastive loss (Hadsell et al., 2006).

3.1.2. Triplet generation

How to generate triplets for training turns out to be vital for learning disentangled representations. A sentence may contain several elements. For example, the sentence underlined in Fig. 1 contains the information about elements e_1 and e_2. For a given element e, if x^p contains more than one element information, it is possible that the learned representation contains irrelevant information. To enforce more effective decoupling of elements, we restrict that x^p is a sentence that is relevant to only one element and x^n is a sentence that is not relevant to e.

For example, in Fig. 1, the sentence highlighted in green is only relevant to e_3, it can be used as x^p for generating triplets for e_3 and used as x^n for generating triplets for other elements, such as e_1. The sentences in black are not relevant to any element, so they can be used as x^n. Note that an element e may involve multiple circumstances. We then further restrict that x^p includes the information of the same circumstance as x, and x^n describes a different circumstance or is not relevant to e. This generation technique is able to substantially decrease the effect of other irrelevant information and generate a large quantity of triplets using a small quantity of labeled data.

3.1.3. Model architecture

Fig. 2 illustrates the architecture of *eVec*. To train the *eVec* model, for a given element e, we first generate triplets (x, x^p, x^n) w.r.t. e, where x is a fact description that contains e, x^p is a sentence describing e and x^n is a sentence not describing e. Each one of x, x^p and x^n consist of a sequence of Chinese words which is independently fed into a word embedding layer. Through the word embedding layer, each word w_i is converted to an embedding $z_i \in \mathbb{R}^m$, where m is the dimension of the embedding. Once converted, the embeddings of all words are stacked in an embedding matrix $Z = [z_1, ..., z_N]$. Many word embedding models, like Glove (Pennington et al., 2014) or Word2vec (Mikolov et al., 2013), can be used for this purpose. The corresponding embedding matrices of x, x^p and x^n, i.e. (Z_x, Z_{x^p}, Z_{x^n}), are then obtained. Next, the embedding matrices of each triplet are fed into three parameter-shared encoder networks to learn higher-order representations $(f_e(x), f_e(x^p), f_e(x^n))$:

$$f_e(x) = Encoder(Z_x), f_e(x^p) = Encoder(Z_{x^p}), f_e(x^n) = Encoder(Z_{x^n}). \quad (3)$$

The three encoder networks are initialized using the same weights and updated with the same gradients. $f_e(\cdot) \in R^h$, where h is the dimension of the representation. Any classical deep neural network can be applied as an encoder network, such as CNN or LSTM. The training objective of the shared encoder network is to minimize the triplet loss (Eq. (2)) to ensure that the similarity between $f_e(x)$ and $f_e(x^p)$ is larger than the similarity between $f_e(x)$ and $f_e(x^n)$. Maximizing the similarity between $f_e(x)$ and $f_e(x^p)$ encourages the encoder network to focus on their common semantic information. Since x^p contains information that is only relevant to the specific element e, the model tends to preserve information about e in $f_e(x)$.

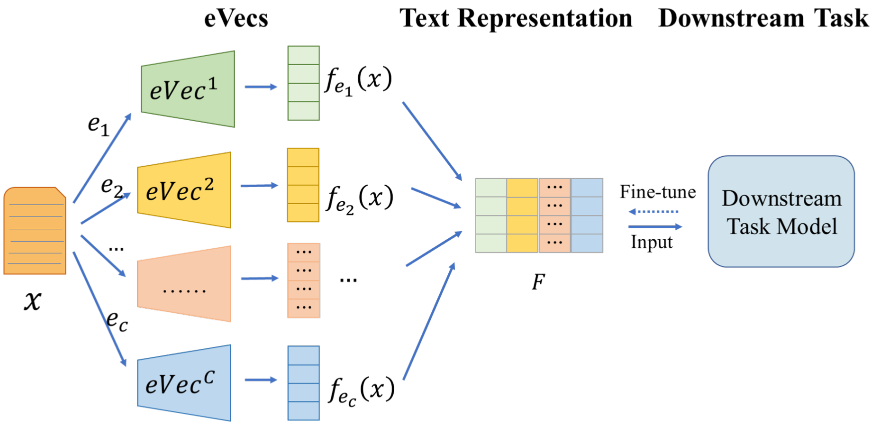


Figure 3: An illustration of the framework *eVecs* for a multi-element task.

3.2. eVecs: a general framework for a multi-element task

In case a task takes multiple elements into consideration, a good text representation should preserve all elements' information and meanwhile well distinguish them. To achieve this, we present *eVecs*, a framework that employs multiple *eVec* models to learn representations w.r.t. multiple elements.

The framework *eVecs* (illustrated in Fig. 3) represents an ensemble model that strives to disentangle elements in a fact description. For a base model *eVec*^{i} and an element e_i, the training triplet generation procedure is the same as described in Section 3.1. A sentence x that describes the element e_i can be used as x or x^p for training *eVec*^{i} and used as x^n for training other base models.

The training procedures of the base models are independent and can be run in parallel. Once all base models are trained, given a fact description x, through base models, the element-related embedding matrix F is obtained as:

$$F = [f_{e_1}(x), ..., f_{e_c}(x)], F \in R^{h*C}, f_{e_i}(x) = eVec^i(x), i = 1, 2, ..., C, \quad (4)$$

where $f_{e_i}(x)$ is the embedding of x w.r.t. e_i, $eVec^i$ is the trained base model w.r.t. e_i and C is the number of elements. Then, for a certain downstream task, there are two options: (1) One can directly apply the representations obtained by the base models to the downstream task; alternatively, (2) the base models can serve as feature extraction layers of the downstream task model and be fine-tuned on the downstream task. The difference between two options is whether the base models participate in the training process for downstream tasks or not.

4. Experiments

4.1. Single-element task: similar case matching

In this section, we conducted an experiment on a synthetic Chinese bankruptcy dataset, to evaluate the effectiveness of the *eVec* model and the triplets generation method in a similar case matching downstream task w.r.t. one specific legal element. Given a triplet containing three fact descriptions (d, d^1, d^2), the task is to determine whether d^1 is more similar to d w.r.t. one element than d^2. If so, then the label of this triplet (d, d^1, d^2) is 1, otherwise it is 0.

4.1.1. Bankruptcy dataset

We collected the civil ruling papers on corporate bankruptcy from the National enterprise bankruptcy information disclosure platform[3] and the Shanghai High People's Court. The civil ruling documents are mainly composed of three components, namely a fact description, relevant law articles and a judgement. In a real application scenario, a judge finds similar cases as references to assist him to make a judgment on the basis of the fact description of a case. Therefore, the analysis was conducted on the fact descriptions.

In the Chinese bankruptcy domain, one important criterion of determining whether a company goes into dissolution is "petition requirements". According to article 2 section 1 of the Enterprise Bankruptcy Law of the People's Republic of China, there are two main circumstances for judging whether a company's liabilities shall be liquidated. Thus, whether two fact descriptions are similar is mainly based on the description of the circumstance of "petition requirements". We selected 84 fact descriptions that are only relevant to one circumstance and ignore those including both circumstances. The petition requirements included in the sentences of each fact description were annotated by legal professionals. For a given sentence, if it contains information related to "petition requirements" , the legal professionals labeled which circumstances of "petition requirements" it was relevant to. Finally, there are 634 sentences relevant to "petition requirements".

We split the selected fact descriptions using a ratio of 4:1 to generate training triplets and test triplets respectively. For training, we generated two groups of triplets, a group of 20,005 s-triplets (sentence-based triplets) and another one of 14,045 f-triplets (fact description-based triplets). As for testing, to resemble a real-world scenario where only f-triplets are available, 2,525 f-triplets were generated. The s-triplets were generated by the method described in Section 3.1, and we

[3] http://pccz.court.gov.cn/pcajxxw/index/xxwsy

restricted that x is a fact description describing one of the circumstances of the petition requirements such that x^p is a sentence describing the same circumstance as the one in x while x^n is a sentence describing the another circumstance. Instead of limiting x^p and x^n to be sentences describing different elements, the f-triplets were generated in a traditional way where x^p is a fact description containing the same element as x and x^n is a fact description relevant to other elements. The number of f-triplets is less than that of s-triplets as there are less fact descriptions than sentences.

4.1.2. Experimental setup

To demonstrate the robustness and effectiveness of the *eVec* model, we considered six basic encoder networks. TextCNN (Kim, 2014) with a window size of the convolution filter in $\{3,4,5\}$ and 128 convolution filters of different width; CNN (LeCun et al., 1998) with one layer of 128 filters and window size of 5; LSTM (Hochreiter & Schmidhuber, 1997) with 128-dimensional hidden states (same below) and an l_2 regularization penalty of 1e-3; LSTM+Attention (Yang et al., 2016); Bi-LSTM (Graves & Schmidhuber, 2005) and Bi-LSTM+Attention (Zhou et al., 2016).

The documents were preprocessed by deleting stop words, punctuation symbols, and words occurring in less than 3 documents, and removing the basic information of a bankrupt company, such as enterprise's address, legal representative and organization code. All documents were tokenized using the jieba[4] toolkit. We only preserved nouns, adjectives and verbs. The embeddings of words were initialized using representations pre-trained on the dataset of Chinese Wikipedia (Li et al., 2018). The size of an embedding is 300. Some professional words in the legal field may not have corresponding representations in the pre-trained model. One straightforward solution is to average the embeddings of all characters in the word to represent it. For example, a word “借条”(receipt for a loan), which is very common in the Chinese bankruptcy application documents but is not among the pre-trained words, can be represented by taking the average of the embeddings of “借” and “条”. To balance the length of sentences and full text, documents were truncated to a fixed length (400).

During the training process, we used Adam (Kingma & Ba, 2015) to optimize the model parameters with a batch size of 32 and a learning rate of 0.001 and the epoch was selected as 20. The margin α was specified as 1 and the initialization

[4] https://github.com/fxsjy/jieba

Table 2: Classification results obtained by the eVec$_b$ and *eVec* models using *six different encoder networks*. eVec$_b$ stands for a "basic" variant of *eVec* that uses fact descriptions to generate triplets.

Encoder network	Accuracy		Precision		Recall		F1-score	
	eVec$_b$	**eVec**	eVec$_b$	**eVec**	eVec$_b$	**eVec**	eVec$_b$	**eVec**
CNN	0.686	**0.871**	0.761	**0.927**	0.660	**0.832**	0.710	**0.877**
TextCNN	0.608	0.799	0.638	0.888	0.602	0.753	0.620	0.815
LSTM	0.472	0.599	0.520	0.757	0.474	0.568	0.496	0.649
LSTM+Att	0.566	0.678	0.604	0.741	0.561	0.658	0.582	0.697
Bi-LSTM	0.577	0.795	0.678	0.893	0.564	0.746	0.616	0.813
Bi-LSTM+Att	0.487	0.819	0.610	0.923	0.490	0.764	0.543	0.836

Table 3: Classification results obtained by the best-performing (CNN-based) *eVec* and *the text representation alternatives*.

Model	Accuracy	Precision	Recall	F1-score
TF-IDF	0.585	0.587	0.585	0.586
Word2vec	0.560	0.594	0.556	0.574
LDA	0.621	0.647	0.614	0.630
LSA	0.570	0.609	0.594	0.601
FastText	0.635	0.616	0.639	0.628
ABAE	0.640	0.617	0.647	0.632
BERT-PLI	0.573	0.523	0.581	0.551
Paraformer	0.601	0.612	0.598	0.605
c-DSSM	0.762	0.857	0.720	0.783
BERT	0.668	0.688	0.661	0.674
SCMKD	0.649	0.682	0.640	0.660
eVec	**0.871**	**0.927**	**0.832**	**0.877**

of other parameters is random. The performance was assessed using accuracy, precision, recall and F1-score.

4.1.3. Effectiveness of the triplet generation strategy

To investigate if the proposed triplet generation strategy is helpful in learning a representation regarding a specific element, we compared the *eVec* model with a basic variant eVec$_b$. The difference is that the *eVec* model is trained with s-triplets and the eVec$_b$ model is trained with f-triplets.

Table 2 shows the classification results obtained by the *eVec* and eVec$_b$ models using 6 different encoder networks. It shows that the *eVec* model made substantial improvements compared to eVec$_b$ w.r.t. different metrics. For example, the *eVec* model achieved 19.4% higher accuracy on average, 21.9% higher precision, 16.2% higher recall and 18.7% higher F1-score. The reason is that x^p and x^n used in eVec$_b$ are composed of the entire texts, which may contain other irrelevant element information. Among the six different encoder networks, CNN achieves the best

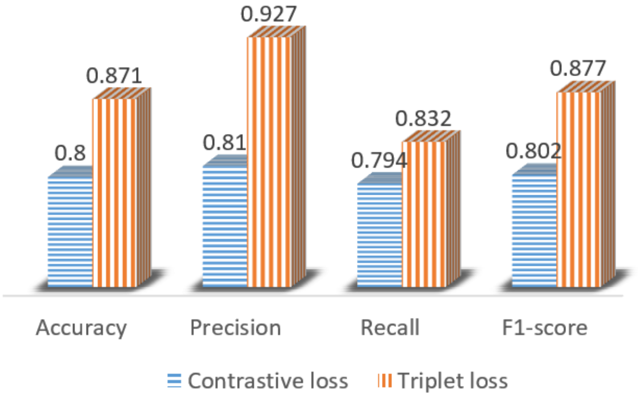


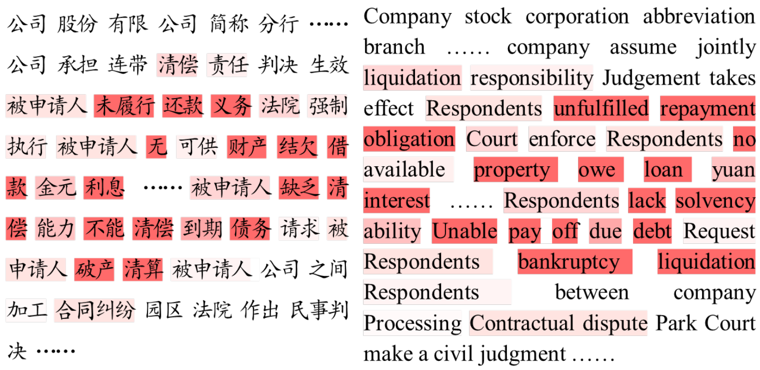
Figure 4: Comparison of loss functions.

performance. This is due to the relatively small amount of selected fact descriptions. Therefore, the complex encoders such as LSTM, are prone to overfitting.

4.1.4. Effectiveness of the eVec model

Next, we compared the *eVec* model with six unsupervised text representation methods including TF-IDF, Word2vec, LDA (Blei et al., 2003), LSA (Dumais, 2004), FastText (Mikolov et al., 2018) and ABAE (He et al., 2017), and five supervised text representation models, BERT-PLI (Shao et al., 2020), Paraformer (Nguyen et al., 2022), c-DSSM (Shen et al., 2014), BERT (Devlin et al., 2019) (initialized with the "bert-base-chinese" pretrained model from transformers[5]) and SCMKD (Peng et al., 2020) (see Table 3). The supervised methods were trained using f-triplets to minimize the triplet loss. According to the results, our method outperforms both unsupervised and supervised methods. Specifically, the *eVec* model achieved 23.1%-31.1% higher accuracy and 24.5%-30.3% higher F1-score compared with the unsupervised methods, and obtained 10.9%-34.2% higher accuracy compared with the supervised methods. It is worth noting that BERT-PLI and Paraformer exhibit lower performance on this task. This is most probably due to the two models having architectures and training processes that might not be well-suited for our specific dataset. Tables 2 and 3 together demonstrate the effectiveness of the *eVec* model.

[5] https://github.com/huggingface/transformers



公司 股份 有限 公司 简称 分行 …… 公司 承担 连带 清偿 责任 判决 生效 被申请人 未履行 还款 义务 法院 强制 执行 被申请人 无 可供 财产 结欠 借款 金元 利息 …… 被申请人 缺乏 清偿 能力 不能 清偿 到期 债务 请求 被申请人 破产 清算 被申请人 公司 之间 加工 合同纠纷 园区 法院 作出 民事判决 ……

Company stock corporation abbreviation branch …… company assume jointly liquidation responsibility Judgement takes effect Respondents unfulfilled repayment obligation Court enforce Respondents no available property owe loan yuan interest …… Respondents lack solvency ability Unable pay off due debt Request Respondents bankruptcy liquidation Respondents between company Processing Contractual dispute Park Court make a civil judgment ……

Figure 5: Visualization of word weights. If a word was assigned a larger weight, the corresponding color is darker.

4.1.5. Effectiveness of the triplet loss

Furthermore, we compared the triplet loss with the contrastive loss (Hadsell et al., 2006). Each s-triplet is decomposed into two pairs, (x, x^p) and (x, x^n), which are the inputs of the contrastive loss. The contrastive loss aims to enforce representations of texts that contain the same element to be similar, or otherwise dissimilar. The triplet loss has a similar goal but it considers the relative similarity between positive text and negative text. For a text, the triplet loss enforces that its similarity to a positive text is higher than its similarity to a negative text by at least a certain threshold. Thus, the triplet loss can better capture the fine-grained differences between elements. For both loss functions, we used CNN as an encoder. Fig. 4 displays the comparison results of using the different loss functions. It indicates that using triplet loss obtains much better results than using contrastive loss.

4.1.6. Visualization of word importance

Fig. 5 shows the weights of words assigned by the LSTM-Attention model. If the weight of a word is larger, then the corresponding color is darker. As expected, the weights of words learned by the *eVec* model strongly correspond to the target element. Words related to "petition requirements", such as "清偿" (solvency), "债务" (debt), "财产" (property), are assigned relatively larger weights. This implies that the *eVec* model is capable of focusing on salient words describing the specific element and the learned text representation could maximally preserve the important information about the target element.

Table 4: The statistics of the three real-world datasets.

Datasets		Loan contracts	Labor dispute	Marriage & family
Training	sentences	5,513	6,698	9,445
	s-triplets	111,467	68,182	96,399
Testing	sentences	1,958	1,574	2,920

Table 5: $F1^{Ave}$-scores obtained by all models for the task of legal identification on the three real-world datasets.

Methods	Loan contracts	Labor dispute	Marriage & family
CNN	0.80±0.02	0.83±0.02	0.87±0.02
TextCNN	0.76±0.02	0.75±0.02	0.85±0.02
CNN+GRU	0.82±0.03	0.85±0.01	0.89±0.02
LSTM	0.75±0.03	0.78±0.02	0.86±0.02
LSTM+Att	0.76±0.02	0.78±0.01	0.86±0.01
Bi-LSTM	0.81±0.02	0.84±0.01	0.89±0.01
Bi-LSTM+Att	0.80±0.01	0.84±0.01	0.88±0.01
ATAE-LSTM	0.81±0.01	0.84±0.01	0.89±0.01
BERT	0.83±0.01	0.88±0.01	0.90±0.01
ELECTRA	0.81±0.01	0.82±0.01	0.88±0.01
Hu et al.	0.81±0.02	0.85±0.01	0.87±0.01
AttentionXML	0.82±0.01	0.86±0.01	0.89±0.01
eVecs	**0.85±0.01**	**0.89±0.01**	**0.91±0.01**

4.2. Multi-element task: legal element identification

We then applied the *eVecs* framework to the legal element identification task. The goal is to disentangle the information of each element in a sentence so that the downstream model can better predict the existence of each element. Given a sentence s, the legal element identification task aims to predict the labels of elements $y = \{y_1, ..., y_c\}$, where $y_l \in \{0, 1\}$ represents the label of an element e_l and c is the total number of element categories.

4.2.1. Real-word datasets

We used a publicly available dataset from the CAIL2019[6] competition. Each record is a sentence of a fact description, and the contained elements in each sentence were labeled by professionals with legal background. A sentence may include the information of multiple elements, and meanwhile there are many sentences that are not associated with any element. The dataset covers three areas, including loan contracts, labor disputes, and marriage and family. For each dataset, we selected the top 10 high-frequency elements as labels. All sentences

[6] http://cail.cipsc.org.cn/

were divided into training and testing sets. Using sentences in the training set, we constructed triplets to train each base *eVec* model w.r.t. an element independently. The detailed statistics of three datasets are listed in Table 4.

4.2.2. Experimental setup

We compared the classification performance of the *eVecs* framework against 11 baselines: CNN, TextCNN, CNN+GRU (Lai et al., 2015), LSTM, LSTM +Attention, Bi-LSTM, Bi-LSTM+Attention, ATAE-LSTM (Wang et al., 2016) with 300-dimensional aspect embeddings and hidden layer of size 128, BERT, ELECTRA (Clark et al., 2020) (itialized with the "chinese-legal-electra-base-discriminator" pretrained model from the "transformers" package[7]), AttentionXML (You et al., 2019), and Hu et al.(Hu et al., 2018) with 128-dimensional hidden states. For baselines that are already used in Section 4.1, the same hyperparameters are set here. We used the same text pre-processing techniques as described in Section 4.1.2. For the baselines, the initial inputs are word embeddings, then the learned representations of sentences are input into a fully connected layer with sigmoid activation, the output of which is the probability of each category. For this task, we did not compare our approach with BERT-PLI and Paraformer, which are specifically designed for processing long case documents, as the text utilized in this task consists of individual sentences. It is important to note that when considering single sentences, these two models have architectures similar to that of BERT.

As for the *eVecs* framework, we used BERT as encoder network for a base model *eVec* to learn text representations for individual elements, as it performed the best among all baselines. The maximum input length of BERT is set to 512, yet most sentences in this task are shorter than 512 characters. Consequently, the input length limitation of BERT does not impact the performance in this task. Here, the CNN is not the best performing encoder, the reason being that the sizes of the datasets in this task are much larger, therefore more complex models (such as BERT) tend to perform better due to their larger modeling capacity. Then, we combined 10 base models using a multi-label classifier. During the training process, the parameters of all base models were fine-tuned. The input of the multi-label classifier was the concatenation of the outputs of all base models. For the classifier architecture, the fully connected layer's dimension was configured to 128 while the other parameters were initialized randomly. We trained all models using Adam

[7] https://github.com/huggingface/transformers

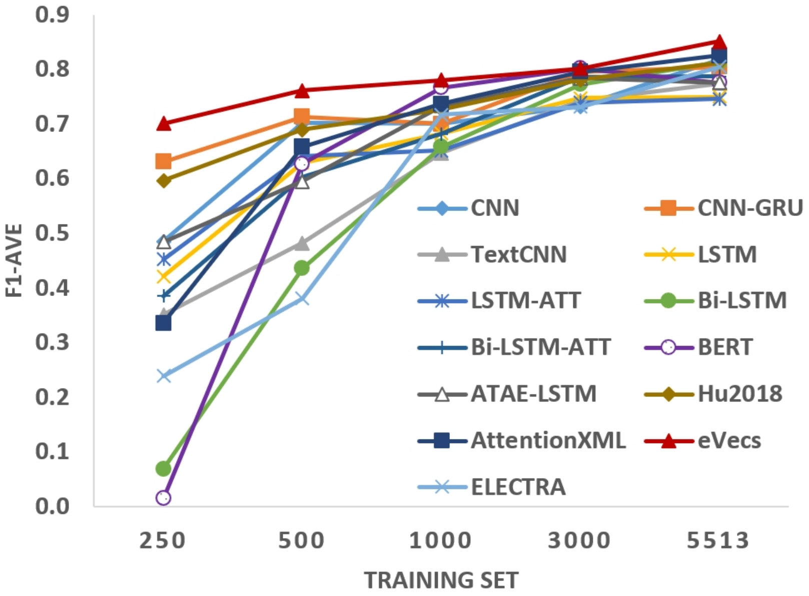
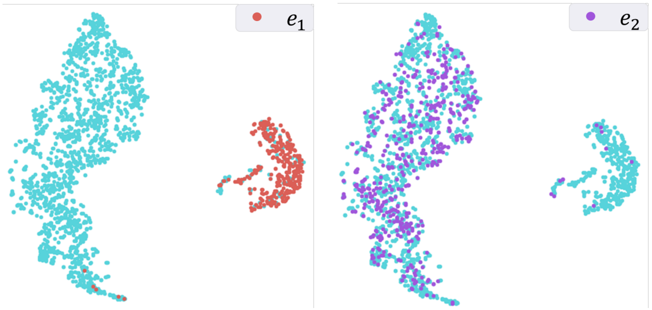


Figure 6: $F1^{ave}$-scores obtained by all models using different amounts of training data from the Loan dataset.

optimizer and the learning rate and batch size were initialized to 0.001 and 32. The performance was evaluated by $F1^{Ave}$-score which is the average of micro-average and macro-average F1-score. Each model was run 10 times and the mean scores and standard deviations were reported.

4.2.3. Effectiveness of the eVecs framework

Table 5 shows the classification results regarding $F1^{ave}$-scores. One can observe that *eVecs* outperforms the alternatives on all three datasets, which manifests the effectiveness of the *eVecs* framework. For example, compared with the other baselines, the *eVecs* framework yielded 2%-9% higher $F1^{Ave}$ on the loan contracts dataset, 1%-14% on the labor dispute dataset, and 1%-6% on the marriage and family dataset, respectively. This proves that the text representations learned by the *eVecs* framework can better capture the properties of each element. We noticed that the lifts made by *eVecs* on the marriage dataset are relatively small compared to other alternatives. The reasons are that the size of labeled marriage data is larger than other two datasets and the number of sentences that are relevant to only one element on the marriage dataset is extremely small. Therefore, when generating triplets for the marriage dataset, we do not restrict that x^p is only relevant to one



(a) (b)

Figure 7: t-SNE plots of the disentangled element in e_1 space. The red dots represent sentences containing element e_1, while the purple dots represent sentences containing element e_2.

element, which essentially decreases the performance of *eVecs*.

4.2.4. Effectiveness w.r.t. data size

We next conducted an experiment to assess the validity of the models using different sizes of training data. We selected 250, 500, 1000, and 3000 sentences from the original Loan dataset to form training datasets of different sizes. Fig. 6 shows the $F1^{ave}$-scores obtained by all models. One can observe that $eVecs$ consistently outperforms all alternatives under different sizes of training data. Especially if the training set is tiny, the superiority of $eVecs$ is more clear. For example, when the data size is 250, the performance of BERT (from Table 5) drops sharply, while our model can maintain stable performance and improve the accuracy by up to 66%. This indicates that $eVecs$ can obtain satisfactory results using just a modest quantity of labeled data, which significantly reduces the cost of labeling.

4.2.5. Latent space visualization

We also examined whether elements are disentangled in the latent space by choosing two elements, assignment of creditor's rights (e_1) and the amount of money borrowed (e_2), from the loan contracts dataset, and retained 1,933 sentences in the test set. We used t-SNE (Maaten & Hinton, 2008) to convert the 128-dimensional learned representations regarding e_1 into a two-dimensional space, and visualize them in Fig. 7. The red dots in Fig. 7(a) represent sentences containing the element e_1, and the purple dots in Fig. 7(b) represent sentences containing the element e_2. The observation is that the red dots in Fig. 7(a) are clustered together and are clearly distant to the rest of the dots, which suggests that the representations of the sentences that contain the same element are more similar in the corresponding element space compared to the ones that include a different element. Moreover, Fig. 7(b) shows that in the latent space of the element e_1, sentences containing e_2 (purple dots) are closely intertwined with sentences that include neither e_1 nor e_2 (blue dots). Note that some sentences containing e_2 (i.e. some of the purple dots) are close to sentences containing e_1. We have separately examined those sentences and found that they are indeed sentences containing both e_1 and e_2. It can be inferred that in the element e_1 space, representations related to e_1 can be well separated from representations not relevant to e_1. Lastly, Fig. 7(a) and Fig. 7(b) together demonstrate the base model's capability of learning disentangled representations.

4.3. Discussion

In this section, we conduct a detailed analysis and discussion of our model. Compared with existing models, the advantages of our model are as follows: 1) It possesses the capability to **learn** representations w.r.t. specific legal elements; 2) It is task-agnostic and can be applied with flexibility across various downstream tasks. 3) The learned representations can be directly applied to or fine-tuned on downstream tasks. There are also some limitations of the proposed model: *1) Relying on labeled elements.* Although the proposed model and its variant rely on labeled elements, the experiments in Section 4.2.4 have demonstrated that even a small amount of labeled data can yield relatively good performance, thus significantly reducing labeling costs. *2) Model complexity.* Training encoders w.r.t. specific elements may initially increase the time complexity of the model, however, once these encoders are trained, they can be directly applied to multiple tasks in the same domain or fine-tuned on a small amount of data. *3) Resource consumption during training and inference.* The resource consumption of our proposed model is contingent on the encoder. Assume that the complexity of the selected encoder is $O(a)$, and there are C elements, then the total complexity of proposed framework is $O(a * C)$. Notably, the training of C element encoders can be conducted in parallel to reduce runtime.

5. Conclusion

In this paper, we proposed the *eVec* model to learn discriminative representations of Chinese legal texts w.r.t. a given element and presented the *eVecs* framework to learn disentangled text representations w.r.t. multiple elements. Experiments on two real-world applications demonstrate that the *eVec* model and the *eVecs* framework can achieve substantial improvements over their alternatives.

CRediT authorship contribution statement

Yingzhi Miao: Conceptualization, Methodology, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing. **Fang Zhou:** Conceptualization, Methodology, Investigation, Writing - Review & Editing, Supervision. **Martin Pavlovski:** Conceptualization, Methodology, Writing - Review & Editing. **Weining Qian:** Supervision, Resources

Acknowledgement

We would like to thank Weixian Zong for conducting the preliminary analysis of the legal element identification task. This work was supported by the National Key R&D Program of China (2018YFC0831904).

References

Bhattacharya, P., Ghosh, K., Pal, A., & Ghosh, S. (2022). Legal case document similarity: You need both network and text. *Information Processing & Management*, *59*(6), 103069. `https://doi.org/10.1016/j.ipm.2022.103069`.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research (JMLR)*, *3*, 993–1022. `https://doi.org/http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993`.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics (EMNLP)* (pp. 2898–2904). Online, Association for Computational Linguistics. `https://doi.org/10.18653/v1/2020.findings-emnlp.261`.

Charmet, T., Cherichi, I., Allain, M., Czerwinska, U., Fouret, A., Sagot, B., & Bawden, R. (2022). Complex Labelling and Similarity Prediction in Legal Texts: Automatic Analysis of France's Court of Cassation Rulings. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC)* (pp. 4754–4766).

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 2172–2180).

Cheng, P., Min, M. R., Shen, D., Malon, C., Zhang, Y., Li, Y., & Carin, L. (2020). Improving Disentangled Text Representation Learning with Information-Theoretic Guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 7530–7541). `https://doi.org/10.18653/v1/2020.acl-main.673`.

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *Proceedings of the International Conference for Learning Representations (ICLR)*.

Denton, E. L., & Birodkar, v. (2017). Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 4414–4423).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)* (pp. 4171–4186). `https://doi.org/10.18653/v1/n19-1423`.

D'Innocente, A., Garg, N., Zhang, Y., Bazzani, L., & Donoser, M. (2021). Localized Triplet Loss for Fine-Grained Fashion Image Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (pp. 3910–3915). `https://doi.org/10.1109/cvprw53098.2021.00435`.

Dumais, S. T. (2004). Latent semantic analysis. *Annual review of information science and technology (ARIST)*, *38*(1), 188–230. `https://doi.org/10.1002/aris.1440380105`.

Fei, H., Li, C., Ji, D., & Li, F. (2022). Mutual disentanglement learning for joint fine-grained sentiment classification and controllable text generation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)* (pp. 1555–1565). `https://doi.org/10.1145/3477495.3532029`.

Feng, Y., Li, C., & Ng, V. (2022). Legal judgment prediction via event extraction with constraints. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 648–664). `https://doi.org/10.18653/v1/2022.acl-long.48`.

Gan, L., Kuang, K., Yang, Y., & Wu, F. (2021). Judgment prediction via injecting legal knowledge into neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (pp. 12866–12874). volume 35. `https://doi.org/10.1609/aaai.v35i14.17522`.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, *18*(5-6), 602–610. `https://doi.org/10.1016/j.neunet.2005.06.042`.

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1735–1742). volume 2. `https://doi.org/10.1109/CVPR.2006.100`.

He, R., Lee, W. S., Ng, H. T., & Dahlmeier, D. (2017). An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 388–397). `https://doi.org/10.18653/v1/p17-1036`.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. `https://doi.org/10.1162/neco.1997.9.8.1735`.

Hu, Z., Li, X., Tu, C., Liu, Z., & Sun, M. (2018). Few-shot charge prediction with discriminative legal attributes. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)* (pp. 487–498).

Jain, S., Banner, E., van de Meent, J.-W., Marshall, I. J., & Wallace, B. C. (2018). Learning disentangled representations of texts with application to biomedical abstracts. In *Proceedings of the 2018 Conference on Empirical Methods Natural Language Processing (EMNLP)* (pp. 4683–4693). `https://doi.org/10.18653/v1/d18-1497`.

John, V., Mou, L., Bahuleyan, H., & Vechtomova, O. (2019). Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the*

57th Annual Meeting of the Association for Computational Linguistics (ACL) (pp. 424–434). https://doi.org/10.18653/v1/p19-1041.

Katz, D. M., Bommarito, M. J., & Blackman, J. (2017). A general approach for predicting the behavior of the Supreme Court of the United States. *PloS One*, *12*(4), 1–18. https://doi.org/10.1371/journal.pone.0174698.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1746–1751). https://doi.org/10.3115/v1/d14-1181.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*.

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent Convolutional Neural Networks for Text Classification. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)* (pp. 2267–2273). https://doi.org/10.1609/aaai.v29i1.9513.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791.

Li, S., Zhao, Z., Hu, R., Li, W., Liu, T., & Du, X. (2018). Analogical Reasoning on Chinese Morphological and Semantic Relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 138–143). https://doi.org/10.18653/v1/p18-2023.

Liu, C.-L., & Hsieh, C.-D. (2006). Exploring phrase-based classification of judicial documents for criminal charges in chinese. In *Proceedings of the 16th International Conference on Foundations of Intelligent Systems (ISMIS)* (pp. 681–690). https://doi.org/10.1007/11875604_75.

Liu, D., Du, W., Li, L., Pan, W., & Ming, Z. (2022). Augmenting Legal Judgment Prediction with Contrastive Case Relations. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)* (pp. 2658–2667).

Long, S., Tu, C., Liu, Z., & Sun, M. (2019). Automatic judgment prediction via legal reading comprehension. In *Proceedings of the 18th Chinese National*

Conference on Computational Linguistics (CCL) (pp. 558–572). https://doi.org/10.1007/978-3-030-32381-3_45.

Ma, L., Zhang, Y., Wang, T., Liu, X., Ye, W., Sun, C., & Zhang, S. (2021). Legal Judgment Prediction with Multi-Stage Case Representation Learning in the Real Court Setting. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)* (pp. 993–1002). https://doi.org/10.1145/3404835.3462945.

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, *9*, 2579–2605.

Mathieu, M. F., Zhao, J. J., Zhao, J., Ramesh, A., Sprechmann, P., & LeCun, Y. (2016). Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 5040–5048).

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2018). Advances in Pre-Training Distributed Word Representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 3111–3119).

Nguyen, H.-T., Phi, M.-K., Ngo, X.-B., Tran, V., Nguyen, L., & Phuong, T. (2022). Attentive deep neural networks for legal document retrieval. *Artificial Intelligence and Law*, 1-30. https://doi.org/10.1007/s10506-022-09341-8.

Peng, D., Yang, J., & Lu, J. (2020). Similar case matching with explicit knowledge-enhanced text representation. *Applied Soft Computing*, *95*, 106514. https://doi.org/10.1016/j.asoc.2020.106514.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). https://doi.org/10.3115/v1/d14-1162.

Shao, Y., Mao, J., Liu, Y., Ma, W., Satoh, K., Zhang, M., & Ma, S. (2020). BERT-PLI: Modeling Paragraph-Level Interactions for Legal Case Retrieval.

In Bessiere, C. (ed.) *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)* (pp. 3501–3507). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2020/484. Main track.

Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014). A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management (CIKM)* (pp. 101–110). https://doi.org/10.1145/2661829.2661935.

Sulea, O.-M., Zampieri, M., Malmasi, S., Vela, M., Dinu, L. P., & Van Genabith, J. (2017). Exploring the use of text classification in the legal domain. In *Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts (ASAIL).*

Tran, V., Nguyen, M. L., & Satoh, K. (2019). Building legal case retrieval systems with lexical matching and summarization using a pre-trained phrase scoring model. In *Proceedings of the 17th International Conference on Artificial Intelligence and Law (ICAIL)* (pp. 275–282). https://doi.org/10.1145/3322640.3326740.

Vuong, Y. T.-H., Bui, Q. M., Nguyen, H.-T., Nguyen, T.-T.-T., Tran, V., Phan, X.-H., Satoh, K., & Nguyen, L.-M. (2022). SM-BERT-CR: A Deep Learning Approach for Case Law Retrieval with Supporting Model. *Artificial Intelligence and Law*, *31*(3), 601–628. https://doi.org/10.1007/s10506-022-09319-6.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., & Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1386–1393). https://doi.org/10.1109/CVPR.2014.180.

Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016). Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing (EMNLP)* (pp. 606–615). https://doi.org/10.18653/v1/d16-1058.

Wang, Z. (2022). Legal Element-oriented Modeling with Multi-view Contrastive Learning for Legal Case Retrieval. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)* (pp. 01–10). https://doi.org/10.1109/ijcnn55064.2022.9892487.

Wiseman, S., Shieber, S., & Rush, A. (2018). Learning Neural Templates for Text Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 3174–3187). https://doi.org/10.18653/v1/d18-1356.

Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., Zhang, T., Han, X., Hu, Z., Wang, H. et al. (2019). Cail2019-scm: A dataset of similar case matching in legal domain. *arXiv preprint arXiv:1911.08962*.

Yan, Y., Zheng, D., Lu, Z., & Song, S. (2017). Event identification as a decision process with non-linear representation of text. *arXiv preprint arXiv:1710.00969*.

Yang, W., Jia, W., Zhou, X., & Luo, Y. (2019). Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 4085–4091). https://doi.org/10.24963/ijcai.2019/567.

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., & Berg-Kirkpatrick, T. (2018). Unsupervised Text Style Transfer using Language Models as Discriminators. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 7287–7298).

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)* (pp. 1480–1489). https://doi.org/10.18653/v1/n16-1174.

Yin, P., Zhou, C., He, J., & Neubig, G. (2018). StructVAE: Tree-structured Latent Variable Models for Semi-supervised Semantic Parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 754–765). https://doi.org/10.18653/v1/p18-1070.

You, R., Zhang, Z., Wang, Z., Dai, S., Mamitsuka, H., & Zhu, S. (2019). Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *In Advances in Neural Information Processing Systems (NIPS)*, 5812–5822.

Zhang, H., Dou, Z., Zhu, Y., & Wen, J.-R. (2023). Contrastive Learning for Legal Judgment Prediction. *ACM Transactions on Information Systems (TOIS)*. `https://doi.org/10.1145/3580489`.

Zhong, H., Zhou, J., Qu, W., Long, Y., & Gu, Y. (2020). An Element-aware Multi-representation Model for Law Article Prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 6663–6668). `https://doi.org/10.18653/v1/2020.emnlp-main.540`.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 207–212). `https://doi.org/10.18653/v1/p16-2034`.

Zhu, A., Yin, Z., Iwana, B. K., Zhou, X., & Xiong, S. (2022). Text Style Transfer based on Multi-factor Disentanglement and Mixture. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)* (pp. 2430–2440). `https://doi.org/10.1145/3503161.3548239`.