Learning from the Best: Rationalizing Prediction by Adversarial Information Calibration

Lei Sha,[1] Oana-Maria Camburu,[1,2] Thomas Lukasiewicz[1,2]
[1]Department of Computer Science, University of Oxford, UK
[2]Alan Turing Institute, London, UK
{lei.sha, oana-maria.camburu, thomas.lukasiewicz}@cs.ox.ac.uk

Abstract

Explaining the predictions of AI models is paramount in safety-critical applications, such as in legal or medical domains. One form of explanation for a prediction is an extractive rationale, i.e., a subset of features of an instance that lead the model to give its prediction on the instance. Previous works on generating extractive rationales usually employ a two-phase model: a selector that selects the most important features (i.e., the rationale) followed by a predictor that makes the prediction based exclusively on the selected features. One disadvantage of these works is that the main signal for learning to select features comes from the comparison of the answers given by the predictor and the ground-truth answers. In this work, we propose to squeeze more information from the predictor via an information calibration method. More precisely, we train two models jointly: one is a typical neural model that solves the task at hand in an accurate but black-box manner, and the other is a selector-predictor model that additionally produces a rationale for its prediction. The first model is used as a guide to the second model. We use an adversarial-based technique to calibrate the information extracted by the two models such that the difference between them is an indicator of the missed or over-selected features. In addition, for natural language tasks, we propose to use a language-model-based regularizer to encourage the extraction of fluent rationales. Experimental results on a sentiment analysis task as well as on three tasks from the legal domain show the effectiveness of our approach to rationale extraction.

1 Introduction

Although deep neural networks have recently been contributing to state-of-the-art advances in various areas (Krizhevsky, Sutskever, and Hinton 2017; Hinton et al. 2012; Sutskever, Vinyals, and Le 2014), such black-box models may not be deemed reliable in situations where safety needs to be guaranteed, such as legal judgment prediction and medical diagnosis. Interpretable deep neural networks are a promising way to increase the reliability of neural models (Sabour, Frosst, and Hinton 2017). To this end, extractive rationales, i.e., subsets of features of instances on

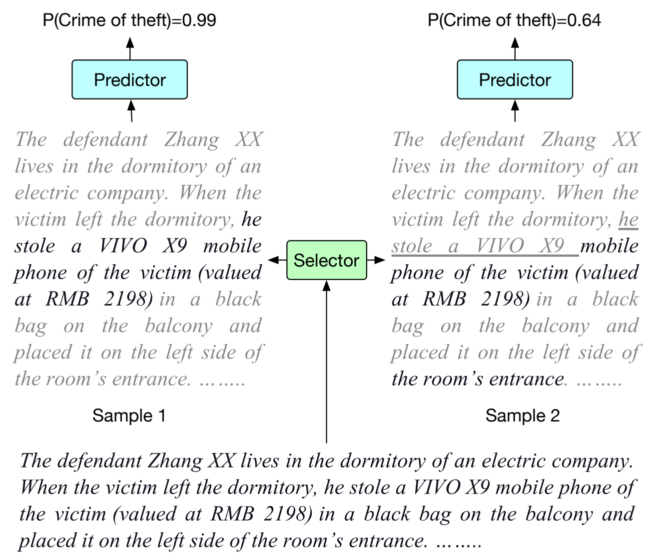

Figure 1: A sample rationale in legal judgement prediction. The human-provided rationale is shown in bold in Sample 1. In Sample 2, the selector missed the key information "he stole a VIVO X9", but the predictor only tells the selector that the whole extracted rationale (in bold) is not so informative, by producing a low probability of the correct crime.

which models rely for their predictions on the instances, can be used as evidence for humans to decide whether or not to trust a predicted result and, more generally, to trust a model.

Previous works mainly use selector-predictor types of neural models to provide extractive rationales, i.e., models composed of two modules: (i) a *selector* that selects a subset of important features, and (ii) a *predictor* that makes a prediction based solely on the selected features. For example, Yoon, Jordon, and van der Schaar (2018) and Lei, Barzilay, and Jaakkola (2016) use a selector network to calculate a selection probability for each token in a sequence, then sample a set of tokens that is exclusively passed to the predictor.

An additional typical desideratum in natural language processing (NLP) tasks is that the selected tokens form a semantically fluent rationale. To achieve this, Lei, Barzilay, and Jaakkola (2016) added a non-differential regularizer that encourages any two adjacent tokens to be simultaneously selected or unselected. Bastings, Aziz, and Titov (2019) further improved the quality of the rationales by using a Hard Kuma regularizer that also encourages any two adjacent tokens to be selected or unselected together.

One drawback of previous works is that the learning signal for both the selector and the predictor comes mainly from comparing the prediction of the selector-predictor model with the ground-truth answer. Therefore, the exploration space to get to the correct rationale is large, decreasing the chances of converging to the optimal rationales and predictions. Moreover, in NLP applications, the regularizers commonly used for achieving fluency of rationales treat all adjacent token pairs in the same way. This often leads to the selection of unnecessary tokens due to their adjacency to informative ones.

In this work, we first propose an alternative method to rationalize the predictions of a neural model. Our method aims to squeeze more information from the predictor in order to guide the selector in selecting the rationales. Our method trains two models: a "guider" model that solves the task at hand in an accurate but black-box manner, and a selector-predictor model that solves the task while also providing rationales. We use an adversarial-based method to encourage the final information vectors generated by the two models to encode the same information. We use an information bottleneck technique in two places: (i) to encourage the features selected by the selector to be the least-but-enough features, and (ii) to encourage the final information vector of the guider model to also contain the least-but-enough information for the prediction. Secondly, we propose using language models as regularizers for rationales in natural language understanding tasks. A language model (LM) regularizer encourages rationales to be fluent subphrases, which means that the rationales are formed by consecutive tokens while avoiding unnecessary tokens to be selected simply due to their adjacency to informative tokens. The effectiveness of our LM-based regularizer is proved by both mathematical derivation and experiments. All the further details are given in the Appendix of the extended (ArXiv) paper.

Our contributions are briefly summarized as follows:

- We introduce an adversarial approach to rationale extraction for neural predictions, which calibrates the information between a guider and a selector-predictor model, such that the selector-predictor model learns to mimic a typical neural model while additionally providing rationales.
- We propose a language-model-based regularizer to encourage the sampled tokens to form fluent rationales.
- We experimentally evaluate our method on a sentiment analysis dataset with ground-truth rationale annotations, and on three tasks of a legal judgement prediction dataset, for which we conducted human evaluations of the extracted rationales. The results show that our method improves over the previous state-of-the-art models in precision and recall of rationale extraction without sacrificing the prediction performance.

2 Approach

Our approach is composed of a selector-predictor architecture, in which we use an information bottleneck technique to restrict the number of selected features, and a guider model, for which we again use the information bottleneck technique to restrict the information in the final feature vector. Then, we use an adversarial method to make the guider model guide the selector to select least-but-enough features. Finally, we use an LM regularizer to make the selected rationale semantically fluent.

2.1 InfoCal: Selector-Predictor-Guider with Information Bottleneck

The high-level architecture of our model, called InfoCal, is shown in Fig. 2. Below, we detail each of its components.

Selector. For a given instance $(\mathbf{x}, y)$, $\mathbf{x}$ is the input with n features $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, and y is the ground-truth corresponding label. The selector network $\mathrm{Sel}(\tilde{\mathbf{z}}_{\mathrm{sym}}|\mathbf{x})$ takes $\mathbf{x}$ as input and outputs $p(\tilde{\mathbf{z}}_{\mathrm{sym}}|\mathbf{x})$, which is a sequence of probabilities $(p_i)_{i=1,\ldots,n}$ representing the probability of choosing each feature x_i as part of the rationale.

Given the sampling probabilities, a subset of features is sampled using the Gumbel softmax (Jang, Gu, and Poole 2016), which provides a differentiable sampling process:

$$u_i \sim U(0,1), \quad g_i = -\log(-\log(u_i)) \tag{1}$$

$$m_i = \frac{\exp((\log(p_i) + g_i)/\tau)}{\sum_j \exp((\log(p_j) + g_j)/\tau)}, \tag{2}$$

where $U(0,1)$ represents the uniform distribution between 0 and 1, and τ is a temperature hyperparameter. Hence, we obtain the sampled mask m_i for each feature x_i, and the vector symbolizing the rationale $\tilde{\mathbf{z}}_{\mathrm{sym}} = (m_1 x_1, \ldots, m_n x_n)$. Thus, $\tilde{\mathbf{z}}_{\mathrm{sym}}$ is the sequence of discrete selected symbolic features forming the rationale.

Predictor. The predictor takes as input the rationale $\tilde{\mathbf{z}}_{\mathrm{sym}}$ given by the selector, and outputs the prediction $\hat{y}_{sp}$. In the selector-predictor part of InfoCal, the input to the predictor is the multiplication of each feature x_i with the sampled mask m_i. The predictor first calculates a dense feature vector $\tilde{\mathbf{z}}_{\mathrm{nero}}$,[1] then uses one feed-forward layer and a softmax layer to calculate the probability distribution over the possible predictions:

$$\tilde{\mathbf{z}}_{\mathrm{nero}} = \mathrm{Pred}(\tilde{\mathbf{z}}_{\mathrm{sym}}) \tag{3}$$

$$p(\hat{y}_{sp}|\tilde{\mathbf{z}}_{\mathrm{sym}}) = \mathrm{Softmax}(W_p \tilde{\mathbf{z}}_{\mathrm{nero}} + b_p). \tag{4}$$

As the input is masked by m_i, the prediction $\hat{y}_{sp}$ is made exclusively based on the features selected by the selector. The loss of the selector-predictor model is the cross-entropy loss:

$$\begin{aligned} L_{sp} &= -\frac{1}{K}\sum_k \log p(y_{\mathrm{sp}}^{(k)}|\mathbf{x}^{(k)}) \\ &= -\frac{1}{K}\sum_k \log \mathbb{E}_{\mathrm{Sel}(\tilde{\mathbf{z}}_{\mathrm{sym}}^{(k)}|\mathbf{x}^{(k)})} p(y_{\mathrm{sp}}^{(k)}|\tilde{\mathbf{z}}_{\mathrm{sym}}^{(k)}) \\ &\leq -\frac{1}{K}\sum_k \mathbb{E}_{\mathrm{Sel}(\tilde{\mathbf{z}}_{\mathrm{sym}}^{(k)}|\mathbf{x}^{(k)})} \log p(y_{\mathrm{sp}}^{(k)}|\tilde{\mathbf{z}}_{\mathrm{sym}}^{(k)}), \end{aligned} \tag{5}$$

[1]Here, "nero" stands for neural feature (i.e., a neural vector representation) as opposed to a symbolic input feature.

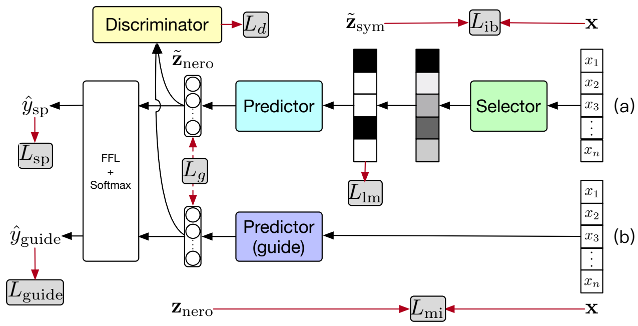

Figure 2: Architecture of InfoCal: the grey round boxes stand for the losses, and the red arrows indicate the data required for the calculation of losses. FFL is an abbreviation for feed-forward layer.

where K represents the size of the training set, the superscript (k) denotes the k-th instance in the training set, and the inequality follows from Jensen's inequality.

Guider. To guide the rationale selection of the selector-predictor model, we train a *guider* model, denoted Pred_G, which receives the full original input $\mathbf{x}$ and transforms it into a dense feature vector $\mathbf{z}_{\text{nero}}$, using the same predictor architecture but different weights, as shown in Fig. 2. We generate the dense feature vector in a variational way, which means that we first generate a Gaussian distribution according to the input $\mathbf{x}$, from which we sample a vector $\mathbf{z}_{\text{nero}}$:

$$h = \text{Pred}_G(\mathbf{x}), \quad \mu = W_m h + b_m, \quad \sigma = W_s h + b_s \tag{6}$$

$$u \sim \mathcal{N}(0, 1), \quad \mathbf{z}_{\text{nero}} = u\sigma + \mu \tag{7}$$

$$p(\hat{y}_{\text{guide}}|\mathbf{z}_{\text{nero}}) = \text{Softmax}(W_p \mathbf{z}_{\text{nero}} + b_p). \tag{8}$$

We use the reparameterization trick of Gaussian distributions to make the sampling process differentiable (Kingma and Welling 2013). Note that we share the parameters W_p and b_p with those in Eq. 4.

The guider model's loss L_{guide} is as follows:

$$\begin{aligned} L_{\text{guide}} &= -\frac{1}{K}\sum_k \log p(y^{(k)}_{\text{guide}}|\mathbf{x}^{(k)}) \\ &\le -\frac{1}{K}\sum_k \mathbb{E}_{p(\mathbf{z}_{\text{nero}}|\mathbf{x}^{(k)})} \log p(y^{(k)}_{\text{guide}}|\mathbf{z}^{(k)}_{\text{nero}}), \end{aligned} \tag{9}$$

where the inequality again follows from Jensen's inequality. The guider and the selector-predictor are trained jointly.

Information Bottleneck. To guide the model to select the least-but-enough information, we employ an information bottleneck technique (Li and Eisner 2019). We aim to minimize $I(\mathbf{x}, \tilde{\mathbf{z}}_{\text{sym}}) - I(\tilde{\mathbf{z}}_{\text{sym}}, y)$[2], where the former term encourages the selection of few features, and the latter term encourages the selection of the necessary features. As $I(\tilde{\mathbf{z}}_{\text{sym}}, y)$ is implemented by L_{sp} (the proof is given in Appendix A.1 in the extended paper), we only need to minimize the mutual information $I(\mathbf{x}, \tilde{\mathbf{z}}_{\text{sym}})$:

$$I(\mathbf{x}, \tilde{\mathbf{z}}_{\text{sym}}) = \mathbb{E}_{\mathbf{x}, \tilde{\mathbf{z}}_{\text{sym}}}\left[\log \frac{p(\tilde{\mathbf{z}}_{\text{sym}}|\mathbf{x})}{p(\tilde{\mathbf{z}}_{\text{sym}})}\right]. \tag{10}$$

However, there is a time-consuming term $p(\tilde{\mathbf{z}}_{\text{sym}}) = \sum_{\mathbf{x}} p(\tilde{\mathbf{z}}_{\text{sym}}|\mathbf{x})p(\mathbf{x})$, which needs to be calculated by a loop over all the instances $\mathbf{x}$ in the training set. Inspired by Li and Eisner (2019), we replace this term with a variational distribution $r_\phi(z)$ and obtain an upper bound of Eq. 10: $I(\mathbf{x}, \tilde{\mathbf{z}}_{\text{sym}}) \le \mathbb{E}_{\mathbf{x}, \tilde{\mathbf{z}}_{\text{sym}}}\left[\log \frac{p(\tilde{\mathbf{z}}_{\text{sym}}|\mathbf{x})}{r_\phi(z)}\right]$. Since $\tilde{\mathbf{z}}_{\text{sym}}$ is a sequence of binary-selected features, we sum up the mutual information term of each element of $\tilde{\mathbf{z}}_{\text{sym}}$ as the information bottleneck loss:

$$L_{\text{ib}} = \sum_i \sum_{\tilde{z}_i} p(\tilde{z}_i|\mathbf{x}) \log \frac{p(\tilde{z}_i|\mathbf{x})}{r_\phi(z_i)}, \tag{11}$$

where $\tilde{z}_i$ represents whether to select the i-th feature: 1 for selected, 0 for not selected.

To encourage $\mathbf{z}_{\text{nero}}$ to contain the least-but-enough information in the guider model, we again use the information bottleneck technique. Here, we minimize $I(\mathbf{x}, \mathbf{z}_{\text{nero}}) - I(\mathbf{z}_{\text{nero}}, y)$. Again, $I(\mathbf{z}_{\text{nero}}, y)$ can be implemented by L_{guide}. Due to the fact that $\mathbf{z}_{\text{nero}}$ is sampled from a Gaussian distribution, the mutual information has a closed-form upper bound:

$$\begin{aligned} L_{\text{mi}} = I(\mathbf{x}, \mathbf{z}_{\text{nero}}) &\le \mathbb{E}_{\mathbf{z}_{\text{nero}}}\left[\log \frac{p(\mathbf{z}_{\text{nero}}|\mathbf{x})}{p(\mathbf{z}_{\text{nero}})}\right] = \\ &= 0.5(\mu^2 + \sigma^2 - 1 - 2\log\sigma). \end{aligned} \tag{12}$$

The derivation is in Appendix A.2 in the extended paper.

2.2 Calibrating Key Features via Adversarial Training

We would like to tell the selector what kind of information is still missing or has been wrongly selected. Since we already use the information bottleneck principal to encourage $\mathbf{z}_{\text{nero}}$ to encode the information from the least-but-enough features, if we also require $\tilde{\mathbf{z}}_{\text{nero}}$ and $\mathbf{z}_{\text{nero}}$ to encode the same information, then we would encourage the selector to select the least-but-enough discrete features. To achieve this, we use an adversarial-based training method. Thus, we employ an additional discriminator neural module, called D, which takes as input either $\tilde{\mathbf{z}}_{\text{nero}}$ or $\mathbf{z}_{\text{nero}}$ and outputs 0 or 1, respectively. The discriminator can be any differentiable neural network. The generator in our model is formed by the selector-predictor that outputs $\tilde{\mathbf{z}}_{\text{nero}}$. The losses associated with the generator and discriminator are:

$$L_d = -\log D(\mathbf{z}_{\text{nero}}) + \log D(\tilde{\mathbf{z}}_{\text{nero}}) \tag{13}$$

$$L_g = -\log D(\tilde{\mathbf{z}}_{\text{nero}}). \tag{14}$$

2.3 Regularizing Rationales with Language Models

For NLP tasks, it is often desirable that a rationale is formed of fluent subphrases (Lei, Barzilay, and Jaakkola 2016). To

[2] $I(a, b) = \int_a \int_b p(a, b) \log \frac{p(a,b)}{p(a)p(b)} = \mathbb{E}_{a,b}[\frac{p(a|b)}{p(a)}]$ denotes the mutual information between the variables a and b.

this end, previous works propose regularizers that bind the adjacent tokens to make them be simultaneously sampled or not. For example, Lei, Barzilay, and Jaakkola (2016) proposed a non-differentiable regularizer trained using REINFORCE (Williams 1992). To make the method differentiable, Bastings, Aziz, and Titov (2019) applied the Kuma-distribution to the regularizer. However, they treat all pairs of adjacent tokens in the same way, although some adjacent tokens have more priority to be bound than others, such as "He stole" or "the victim" rather than ". He" or ") in" in Fig. 1.

We propose a novel differentiable regularizer for extractive rationales that is based on a pre-trained language model, thus encouraging both consecutiveness and fluency of tokens in the extracted rationale. The LM-based regularizer is implemented as follows:

$$L_{\text{lm}} = -\sum_i m_{i-1} \log p_{lm}(m_i x_i | \mathbf{x}_{<i}), \tag{15}$$

where the m_i's are the masks obtained in Eq. 2. Note that non-selected tokens are masked instead of deleted in this regularizer. The language model can have any architecture.

First, we note that L_{lm} is differentiable. Secondly, the following theorem guarantees that L_{lm} encourages consecutiveness of selected tokens.

Theorem 1. *If the following is satisfied for all i, j:*

- $m'_i < \epsilon \ll 1 - \epsilon < m_i,\ 0 < \epsilon < 1$, *and*
- $\left|p(m'_i x_i | x_{<i}) - p(m'_j x_j | x_{<j})\right| < \epsilon$,

then the following two inequalities hold:
(1) $L_{lm}(\ldots, m_k, \ldots, m'_n) < L_{lm}(\ldots, m'_k, \ldots, m_n)$.
(2) $L_{lm}(m_1, \ldots, m'_k, \ldots) > L_{lm}(m'_1, \ldots, m_k, \ldots)$.

The theorem says that for the same number of selected tokens, if they are consecutive, then they will get a lower L_{lm} value. Its proof is given in Appendix A.3 in the extended paper. The pre-training procedure of the language model is shown in Appendix C in the extended paper.

2.4 Training and Inference

The total loss function of our model, which takes the generator's role in adversarial training, is shown in Eq. 17. The adversarial-related losses are denoted by L_{adv}. The discriminator is trained by L_d from Eq. 13.

$$L_{\text{adv}} = \lambda_g L_g + L_{guide} + \lambda_{mi} L_{\text{mi}} \tag{16}$$

$$J_{\text{total}} = L_{sp} + \lambda_{ib} L_{\text{ib}} + L_{\text{adv}} + \lambda_{lm} L_{\text{lm}}, \tag{17}$$

where $\lambda_{ib}, \lambda_g, \lambda_{mi}$, and λ_{lm} are hyperparameters.

At training time, we optimize the generator loss J_{total} and discriminator loss L_d alternately until convergence. The detailed algorithm for training is given in Appendix D in the extended paper. At inference time, we run the selector-predictor model to obtain the prediction and the rationale $\tilde{\mathbf{z}}_{\text{sym}}$.

3 Experiments

We performed experiments on two NLP applications: multi-aspect sentiment analysis and legal judgement prediction.

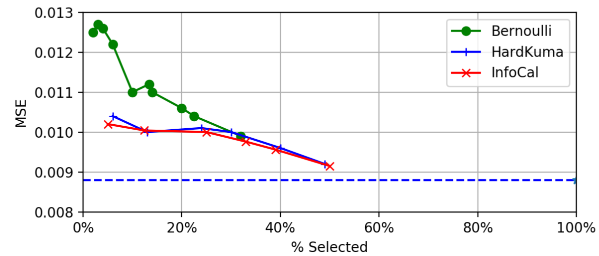

Figure 3: MSE of all aspects of BeerAdvocate. The blue dashed line represents the full-text baseline (all tokens are selected).

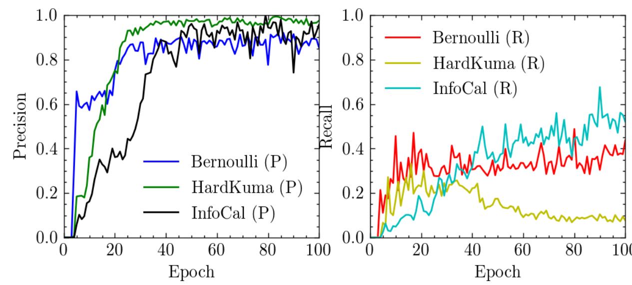

Figure 4: The precision (left) and recall (right) for rationales on the smell aspect of the BeerAdvocate test set.

3.1 Beer Reviews

Data. To provide a quantitative analysis for the extracted rationales, we use the BeerAdvocate[3] dataset (McAuley, Leskovec, and Jurafsky 2012). This dataset contains instances of human-written multi-aspect reviews on beers. Similarly to Lei, Barzilay, and Jaakkola (2016), we consider the following three aspects: appearance, smell, and palate. McAuley, Leskovec, and Jurafsky (2012) provide manually annotated rationales for 994 reviews for all aspects, which we use as test set. The detailed data preprocessing and experimental settings are given in Appendix B in the extended paper.

Evaluation Metrics and Baselines. For the evaluation of the selected tokens as rationales, we use precision, recall, and F1-score. Typically, precision is defined as the percentage of selected tokens that also belong to the human-annotated rationale. Recall is the percentage of human-annotated rationale tokens that are selected by our model. The predictions made by the selected rationale tokens are evaluated using the mean-square error (MSE).

We compare our method with the following baselines:

- Attention (Lei, Barzilay, and Jaakkola 2016): This method calculates attention scores over the tokens and selects top-k percent tokens as the rationale.
- Bernoulli (Lei, Barzilay, and Jaakkola 2016): This method uses a selector network to calculate a Bernoulli distribution for each token, and then samples the tokens from the

[3]https://www.beeradvocate.com/

Method	Appearance				Smell				Palate			
	P	R	F	% selected	P	R	F	% selected	P	R	F	% selected
Attention	80.6	35.6	49.4	13	88.4	20.6	33.4	7	65.3	35.8	46.2	7
Bernoulli	96.3	56.5	71.2	14	95.1	38.2	54.5	7	80.2	53.6	64.3	7
HardKuma	98.1	65.1	78.3	13	**96.8**	31.5	47.5	7	**89.8**	48.6	63.1	7
InfoCal	**98.5**	**73.2**	**84.0**	13	95.6	**45.6**	**61.7**	7	89.6	**59.8**	**71.7**	7
InfoCal(HK)	97.9	71.7	82.8	13	94.8	42.3	58.5	7	89.4	56.9	69.5	7
InfoCal$-L_{\text{adv}}$	97.3	67.8	79.9	13	94.3	34.5	50.5	7	89.6	51.2	65.2	7
InfoCal$-L_{\text{lm}}$	79.8	54.9	65.0	13	87.1	32.3	47.1	7	83.1	47.4	60.4	7

Table 1: Precision, recall, and F1-score of selected rationales for the three aspects of BeerAdvocate. In bold, the best performance. "% selected" means the average percentage of tokens selected out of the total number of tokens per instance.

distributions as the rationale.

- HardKuma (Bastings, Aziz, and Titov 2019): This method replaces the Bernoulli distribution by a Kuma distribution to facilitate differentiability.

The details of the choice of neural architecture for each module of our model, as well as the training setup are given in Appendix B in the extended paper.

Results. The rationale extraction performances are shown in Table 1. The precision values for the baselines are directly taken from (Bastings, Aziz, and Titov 2019). We use their source code for the Bernoulli[4] and HardKuma[5] baselines. We trained these baseline for 50 epochs and selected the models with the best recall on the dev set when the precision was equal or larger than the reported dev precision. For fair comparison, we used the same stopping criteria for InfoCal (for which we fixed a threshold for the precision at 2% lower than the previous state-of-the-art).

We also conducted ablation studies: (1) we removed the adversarial loss and report the results in the line InfoCal$-L_{\text{adv}}$, and (2) we removed the LM regularizer and report the results in the line InfoCal$-L_{\text{lm}}$.

In Table 1, we see that, although Bernoulli and HardKuma achieve very high precisions, their recall scores are significantly low. In comparison, our method InfoCal significantly outperforms the previous methods in the recall scores for all the three aspects of the BeerAdvocate dataset (we performed Student's t-test, $p < 0.01$). Also, all the three F-scores of InfoCal are a new state-of-the-art performance.

In the ablation studies, we see that when we remove the adversarial information calibrating structure, namely, for InfoCal$-L_{\text{adv}}$, the recall scores decrease significantly in all the three aspects. This shows that our guider model is critical for the increased performance. Moreover, when we remove the LM regularizer, we find a significant drop in both precision and recall, in the line InfoCal$-L_{\text{lm}}$. This highlights the importance of semantical fluency of rationales, which are encouraged by our LM regularizer.

We also replace the LM regularizer with the regularizer used in the HardKuma method with all the other parts of the model unchanged, denoted InfoCal(HK) in Table 1. We found that the recall and F-score of InfoCal outperforms InfoCal(HK), which shows the effectiveness of our LM regularizer.

We further show the relation between a model's performance on predicting the final answer and the rationale selection percentage (which is determined by the model) in Fig. 3, as well as the relation between precision/recall and training epochs in Fig. 4. The rationale selection percentage is influenced by λ_{ib}. According to Fig. 3, our method InfoCal achieves a similar prediction performance compared to previous works, and does slightly better than HardKuma for some selection percentages. Fig. 4 shows the changes in precision and recall with training epochs. We can see that our model achieves a similar precision after several training epochs, while significantly outperforming the previous methods in recall, which proves the effectiveness of our proposed method.

3.2 Legal Judgement Prediction

Datasets and Preprocessing. We use the CAIL2018 dataset[6] (Zhong et al. 2018) for three tasks on legal judgment prediction. The dataset consists of criminal cases published by the Supreme People's Court of China.[7] To be consistent with previous works, we used two versions of CAIL2018, namely, CAIL-small (the exercise stage data) and CAIL-big (the first stage data).

The instances in CAIL2018 consist of a *fact description* and three kinds of annotations: *applicable law articles*, *charges*, and *the penalty terms*. Therefore, our three tasks on this dataset consist of predicting (1) law articles, (2) charges, and (3) terms of penalty according to the given fact description. The detailed experimental settings are given in Appendix B in the extended paper.

Overall Performance. We again compare our method with the Bernoulli (Lei, Barzilay, and Jaakkola 2016) and the HardKuma (Bastings, Aziz, and Titov 2019) methods on rationale extraction. These two methods are both single-task models, which means that we train a model separately for each task. We also compare our method with three multi-task methods listed as follows:

- FLA (Luo et al. 2017) uses an attention mechanism to

[4]https://github.com/taolei87/rcnn

[5]https://github.com/bastings/interpretable_predictions

[6]https://cail.oss-cn-qingdao.aliyuncs.com/CAIL2018_ALL_DATA.zip

[7]http://cail.cipsc.org.cn/index.html

Small	Tasks	Law Articles					Charges					Terms of Penalty				
	Metrics	Acc	MP	MR	F1	%S	Acc	MP	MR	F1	%S	Acc	MP	MR	F1	%S
Single	Bernoulli	0.812	0.726	0.765	0.756	100	0.810	0.788	0.760	0.777	100	0.331	0.323	0.297	0.306	100
	Bernoulli	0.755	0.701	0.737	0.728	14	0.761	0.753	0.739	0.754	14	0.323	0.308	0.265	0.278	30
	HardKuma	0.807	0.704	0.757	0.739	100	0.811	0.776	0.763	0.776	100	0.345	0.355	0.307	0.319	100
	HardKuma	0.783	0.706	0.735	0.729	14	0.778	0.757	0.714	0.736	14	0.340	0.328	0.296	0.309	30
	InfoCal	0.834	0.744	0.776	0.786	14	0.849	0.817	0.798	0.813	14	0.358	0.372	0.335	0.337	30
	InfoCal $-L_{\text{adv}}$	0.826	0.739	0.774	0.777	14	0.845	0.804	0.781	0.797	14	0.351	0.374	0.329	0.330	30
	InfoCal $-L_{\text{adv}}-L_{\text{ib}}$	**0.841**	**0.759**	**0.785**	**0.793**	100	**0.850**	**0.820**	**0.801**	**0.814**	100	**0.368**	**0.378**	**0.341**	**0.346**	100
	InfoCal $-L_{\text{lm}}$	0.822	0.723	0.768	0.773	14	0.843	0.796	0.770	0.772	14	0.347	0.361	0.318	0.320	30
Multi	FLA	0.803	0.724	0.720	0.714	–	0.767	0.758	0.738	0.732	–	0.371	0.310	0.300	0.299	–
	TOPJUDGE	0.872	0.819	0.808	0.800	–	0.871	0.864	0.851	0.846	–	0.380	0.350	0.353	0.346	–
	MPBFN-WCA	0.883	0.832	0.824	0.822	–	0.887	0.875	0.857	0.859	–	0.414	0.406	0.369	0.392	–
Big	Tasks	Law Articles					Charges					Terms of Penalty				
	Metrics	Acc	MP	MR	F1	%S	Acc	MP	MR	F1	%S	Acc	MP	MR	F1	%S
Single	Bernoulli	0.876	0.636	0.388	0.625	100	0.857	0.643	0.410	0.569	100	0.509	0.511	0.304	0.312	100
	Bernoulli	0.857	0.632	0.374	0.621	14	0.848	0.635	0.402	0.543	14	0.496	0.505	0.289	0.306	30
	HardKuma	0.907	0.664	0.397	0.627	100	0.907	0.689	0.438	0.608	100	0.555	0.547	0.335	0.356	100
	HardKuma	0.876	0.645	0.384	0.609	14	0.892	0.676	0.425	0.587	14	0.534	0.535	0.310	0.334	30
	InfoCal	0.956	0.852	0.742	**0.805**	20	0.955	0.868	**0.788**	**0.820**	20	0.556	**0.519**	0.362	0.372	30
	InfoCal $-L_{\text{adv}}$	0.953	0.844	0.711	0.782	20	0.954	0.857	0.772	0.806	20	0.552	0.490	0.353	0.356	30
	InfoCal $-L_{\text{adv}}-L_{\text{ib}}$	**0.959**	**0.862**	**0.751**	0.791	100	**0.957**	**0.878**	0.776	0.807	100	**0.584**	**0.519**	**0.411**	**0.427**	30
	InfoCal $-L_{\text{lm}}$	0.953	0.851	0.730	0.775	20	0.950	0.857	0.756	0.789	20	0.563	0.486	0.374	0.367	30
Multi	FLA	0.942	0.763	0.695	0.746	–	0.931	0.798	0.747	0.780	–	0.531	0.437	0.331	0.370	–
	TOPJUDGE	0.963	0.870	0.778	0.802	–	0.960	0.906	0.824	0.853	–	0.569	0.480	0.398	0.426	–
	MPBFN-WCA	0.978	0.872	0.789	0.820	–	0.977	0.914	0.836	0.867	–	0.604	0.534	0.430	0.464	–

Table 2: The overall performance on the CAIL2018 dataset (Small and Big). The results from previous works are directly quoted from Yang et al. (2019), because we share the same experimental settings, and hence we can make direct comparisons. %S represents the selection percentage (which is determined by the model). "Single" represents single-task models, "Multi" represents multi-task models. The best performance is in bold. The red numbers mean that they are less than the best performance by no more than 0.01. The underlined numbers are the state-of-the-art performances, all of which are obtained by multi-task models.

capture the interaction between fact descriptions and applicable law articles.

- TOPJUDGE (Zhong et al. 2018) uses a topological architecture to link different legal prediction tasks together, including the prediction of law articles, charges, and terms of penalty.
- MPBFN-WCA (Yang et al. 2019) uses a backward verification to verify upstream tasks given the results of downstream tasks.

The results are listed in Table 2.

On CAIL-small, we observe that it is more difficult for the single-task models to outperform multi-task methods. This is likely due to the fact that the tasks are related, and learning them together can help a model to achieve better performance on each task separately. After removing the restriction of the information bottleneck, InfoCal$-L_{\text{adv}}-L_{\text{ib}}$ achieves the best performance in all tasks, however, it selects all the tokens in the review. When we restrict the number of selected tokens to 14% (by tuning the hyperparameter λ_{ib}), InfoCal (in red) only slightly drops in all evaluation metrics, and it already outperforms Bernoulli and HardKuma, even if they have used all tokens. This means that the 14% selected tokens are very important to the predictions. We observe a similar phenomenon for CAIL-big. Specifically, InfoCal outperforms InfoCal$-L_{\text{adv}}-L_{\text{ib}}$ in some evaluation metrics, such as the F1-score of law article prediction and charge prediction tasks.

Rationales. The CAIL2018 dataset does not contain annotations of rationales. Therefore, we conducted human evaluation for the extracted rationales. Due to limited budget and resources, we sampled 300 examples for each task. We randomly shuffled the rationales for each task and asked six undergraduate students from Peking University to evaluate them. The human evaluation is based on three metrics: usefulness (U), completeness (C), and fluency (F); each scored from 1 (lowest) to 5. The scoring standard for human annotators is given in Appendix E in the extended paper.

The human evaluation results are shown in Table 3. We can see that our proposed method outperforms previous methods in all metrics. Our inter-rater agreement is acceptable by Krippendorff's rule (2004), which is shown in Table 3.

A sample case of extracted rationales in legal judgement is shown in Fig. 5. We observe that our method selects all the useful information for the charge prediction task, and the selected rationales are formed of continuous and fluent sub-phrases.

4 Related Work

Explainability is currently a key bottleneck of deep-learning-based approaches. The model proposed in this work belongs to the class of self-explanatory models, which contain an explainable structure in the model architecture, thus providing explanations for their predictions. Self-explanatory models can use different types of explanations, such as feature-based explanations (Lei, Barzilay, and Jaakkola

	Law			Charges			ToP		
	U	C	F	U	C	F	U	C	F
Bernoulli	4.71	2.46	3.45	3.67	2.35	3.45	3.35	2.76	3.55
HardKuma	4.65	3.21	3.78	4.01	3.26	3.44	3.84	2.97	3.76
InfoCal	**4.72**	**3.78**	**4.02**	**4.65**	**3.89**	**4.23**	**4.21**	**3.43**	**3.97**
α	0.81	0.79	0.83	0.92	0.85	0.87	0.82	0.83	0.94

Table 3: Human evaluation on the CAIL2018 dataset. "ToP" is the abbreviation of "Terms of Penalty". The metrics are: usefulness (U), completeness (C), and fluency (F), each scored from 1 to 5. Best performance is in bold. α represents Krippendorff's alpha values.

> The People's Procuratorate of Yongshun County alleged that on January 11, 2014, the defendant Li XX and Peng XX (a separate case dealt with) forcibly had sexual relations with the victim Zou XX in a room of Xindu Hotel in Yongshun County . In this regard, the public prosecution agency cited the following evidence: capture history, household registration certificate, call list, description of the situation; identification transcripts; on-site inspection transcripts and on-site photos; physical evidence inspection reports and physical evidence identification documents; witnesses Liu A, Liu B, Testimony of Liu C, Zou XX, Du XX; confession and defense of defendant Li XX; audio-visual materials. The court held that the defendant Li XX used violence and verbal threats with others to forcibly have sexual relations with the victim Zou XX in the Xindu Hotel room in Yongshun County. His behavior has violated the Item (4) of the Criminal Law of the PRC, the facts of the crime are clear, and the evidence is reliable and sufficient, and the criminal responsibility should be investigated for the crime of × ×. In the joint crime, the defendant Li XX played the main role and was the principal offender.....

Figure 5: An example of extracted rationale for charge prediction. The correct charge is "Rape". The original fact description is in Chinese, we have translated it to English. It is easy to see that the extracted rationales are very helpful in making the charge prediction.

2016; Yoon, Jordon, and van der Schaar 2018; Chen et al. 2018; Yu et al. 2019; Carton, Mei, and Resnick 2018) and natural language explanations (Hendricks et al. 2016; Camburu et al. 2018; Park et al. 2018; Kim et al. 2018). Our model uses feature-based explanations.

Self-explanatory models with feature-based explanations can be further divided into two branches. The first branch is formed of representation-interpretable approaches, which map specific features into latent spaces and then use the latent variables to control the outcomes of the model, such as disentangling methods (Chen et al. 2016; Sha and Lukasiewicz 2021), information bottleneck methods (Tishby, Pereira, and Bialek 2000), and constrained generation (Sha 2020). The second branch consists of architecture-interpretable models, such as attention-based models (Zhang et al. 2018; Sha et al. 2016, 2018a,b; Liu et al. 2018), neural Turing machines (Collier and Beel 2018; Xia et al. 2017; Sha et al. 2020), capsule networks (Sabour, Frosst, and Hinton 2017), and energy-based models (Grathwohl et al. 2019). Among them, attention-based models have an important extension, that of sparse feature learning, which implies learning to extract a subset of features that are most informative for each example. Most of the sparse feature learning methods use a selector-predictor architecture. Among them, L2X (Chen et al. 2018) and INVASE (Yoon, Jordon, and van der Schaar 2018) make use of information theories for feature selection, while CAR (Chang et al. 2019) extracts useful features in a game-theoretic approach.

In addition, rationale extraction for NLP usually raises one desideratum for the extracted subset of tokens: rationales need to be fluent subphrases instead of separate tokens. To this end, Lei, Barzilay, and Jaakkola (2016) proposed a non-differentiable regularizer to encourage selected tokens to be consecutive, which can be optimized by REINFORCE-style methods (Williams 1992). Bastings, Aziz, and Titov (2019) proposed a differentiable regularizer using the Hard Kumaraswamy distribution; however, this still does not consider the difference in the importance of different adjacent token pairs.

Our adversarial calibration method is inspired by distilling methods (Hinton, Vinyals, and Dean 2015). Distilling methods are usually applied to compress large models into small models while keeping a comparable performance. For example, TinyBERT (Jiao et al. 2019) is a distillation of BERT (Devlin et al. 2019). Our method is different from distilling methods, because we calibrate the final feature vector instead of the softmax prediction.

The information bottleneck (IB) theory is an important basic theory of neural networks (Tishby, Pereira, and Bialek 2000). It originated in information theory and has been widely used as a theoretical framework in analyzing deep neural networks (Tishby and Zaslavsky 2015). For example, Li and Eisner (2019) used IB to compress word embeddings in order to make them contain only specialized information, which leads to a much better performance in parsing tasks.

Adversarial methods, which had been widely applied in image generation (Chen et al. 2016) and text generation (Yu et al. 2017), usually have a discriminator and a generator. The discriminator receives pairs of instances from the real distribution and from the distribution generated by the generator, and it is trained to differentiate between the two. The generator is trained to fool the discriminator (Goodfellow et al. 2014). Our information calibration method generates a dense feature vector using selected symbolic features, and the discriminator is used for measuring the calibration extent.

5 Summary and Outlook

In this work, we proposed a novel method to extract rationales for neural predictions. Our method uses an adversarial-based technique to make a selector-predictor model learn from a guider model. In addition, we proposed a novel regularizer based on language models, which makes the extracted rationales semantically fluent. The experimental results showed that our method improves the selection of rationales by a large margin.

As future work, the main architecture of our model can be directly applied to other domains, e.g., images or tabular data. However, it remains an open question what would be a good regularizer for these domains.

Acknowledgments

This work was supported by the EPSRC grant "Unlocking the Potential of AI for English Law", a JP Morgan PhD Fellowship, the Alan Turing Institute under the EPSRC grant EP/N510129/1, and the AXA Research Fund. We also acknowledge the use of Oxford's Advanced Research Computing (ARC) facility, of the EPSRC-funded Tier 2 facility JADE (EP/P020275/1), and of GPU computing support by Scan Computers International Ltd.

References

Bastings, J.; Aziz, W.; and Titov, I. 2019. Interpretable Neural Predictions with Differentiable Binary Variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2963–2977.

Camburu, O.; Rocktäschel, T.; Lukasiewicz, T.; and Blunsom, P. 2018. e-SNLI: Natural Language Inference with Natural Language Explanations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018 (NeurIPS 2018)*, 9560–9572.

Carton, S.; Mei, Q.; and Resnick, P. 2018. Extractive Adversarial Networks: High-Recall Explanations for Identifying Personal Attacks in Social Media Posts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3497–3507. Association for Computational Linguistics.

Chang, S.; Zhang, Y.; Yu, M.; and Jaakkola, T. 2019. A Game Theoretic Approach to Class-wise Selective Rationalization. In *Advances in Neural Information Processing Systems*, 10055–10065.

Chen, J.; Song, L.; Wainwright, M.; and Jordan, M. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In *Proceedings of the International Conference on Machine Learning*, 883–892.

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

Collier, M.; and Beel, J. 2018. Implementing Neural Turing Machines. In *Proceedings of the International Conference on Artificial Neural Networks*, 94–104. Springer.

Devlin, J.; Chang, M.-W.; Lee, K.; and 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, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, 2672–2680.

Grathwohl, W.; Wang, K.-C.; Jacobsen, J.-H.; Duvenaud, D.; Norouzi, M.; and Swersky, K. 2019. Your Classifier is Secretly an Energy Based Model and You Should Treat It Like One. *arXiv preprint arXiv:1912.03263* .

Hendricks, L. A.; Akata, Z.; Rohrbach, M.; Donahue, J.; Schiele, B.; and Darrell, T. 2016. Generating Visual Explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19. Springer.

Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Kingsbury, B.; and Sainath, T. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine* 29: 82–97.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531* .

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical Reparameterization with Gumbel-softmax. *arXiv preprint arXiv:1611.01144* .

Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; and Liu, Q. 2019. TinyBERT: Distilling BERT for Natural Language Understanding. *arXiv preprint arXiv:1909.10351* .

Kim, J.; Rohrbach, A.; Darrell, T.; Canny, J. F.; and Akata, Z. 2018. Textual Explanations for Self-Driving Vehicles. *Proceedings of the European Conference on Computer Vision (ECCV)* .

Kingma, D. P.; and Welling, M. 2013. Auto-encoding Variational Bayes. *arXiv preprint arXiv:1312.6114* .

Krippendorff, K. 2004. Content Analysis: An Introduction to Its Methodology Thousand Oaks. *Calif.: Sage* .

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. Imagenet Classification with Deep Convolutional Neural Networks. *Communications of the ACM* 60(6): 84–90.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 107–117.

Li, X. L.; and Eisner, J. 2019. Specializing Word Embeddings (for Parsing) by Information Bottleneck. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2744–2754. Association for Computational Linguistics.

Liu, T.; Wang, K.; Sha, L.; Chang, B.; and Sui, Z. 2018. Table-to-text Generation by Structure-aware Seq2seq Learning. *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* .

Luo, B.; Feng, Y.; Xu, J.; Zhang, X.; and Zhao, D. 2017. Learning to Predict Charges for Criminal Cases with Legal Basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2727–2736. Association for Computational Linguistics.

McAuley, J.; Leskovec, J.; and Jurafsky, D. 2012. Learning Attitudes and Attributes From Multi-aspect Reviews. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, 1020–1025. IEEE.

Park, D. H.; Hendricks, L. A.; Akata, Z.; Rohrbach, A.; Schiele, B.; Darrell, T.; and Rohrbach, M. 2018. Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sabour, S.; Frosst, N.; and Hinton, G. E. 2017. Dynamic Routing Between Capsules. In *Advances in Neural Information Processing Systems*, 3856–3866.

Sha, L. 2020. Gradient-guided Unsupervised Lexically Constrained Text Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8692–8703. Association for Computational Linguistics.

Sha, L.; Chang, B.; Sui, Z.; and Li, S. 2016. Reading and Thinking: Re-read LSTM Unit for Textual Entailment Recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2870–2879.

Sha, L.; and Lukasiewicz, T. 2021. Multi-type Disentanglement without Adversarial Training. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

Sha, L.; Mou, L.; Liu, T.; Poupart, P.; Li, S.; Chang, B.; and Sui, Z. 2018a. Order-Planning Neural Text Generation From Structured Data. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Sha, L.; Shi, C.; Chen, Q.; Zhang, L.; and Wang, H. 2020. Estimate Minimum Operation Steps via Memory-based Recurrent Calculation Network. In *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE.

Sha, L.; Zhang, X.; Qian, F.; Chang, B.; and Sui, Z. 2018b. A Multi-view Fusion Neural Network for Answer Selection. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. *CoRR* abs/1409.3215.

Tishby, N.; Pereira, F. C.; and Bialek, W. 2000. The Information Bottleneck Method. *arXiv preprint physics/0004057* .

Tishby, N.; and Zaslavsky, N. 2015. Deep Learning and The Information Bottleneck Principle. In *Proceedings of the 2015 IEEE Information Theory Workshop (ITW)*, 1–5. IEEE.

Williams, R. J. 1992. Simple Statistical Gradient-following Algorithms for Connectionist Reinforcement Learning. *Machine Learning* 8(3-4): 229–256.

Xia, Q.; Sha, L.; Chang, B.; and Sui, Z. 2017. A Progressive Learning Approach to Chinese SRL Using Heterogeneous Data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2069–2077. Association for Computational Linguistics.

Yang, W.; Jia, W.; Zhou, X.; and Luo, Y. 2019. Legal Judgment Prediction via Multi-perspective Bi-feedback Network. *arXiv preprint arXiv:1905.03969* .

Yoon, J.; Jordon, J.; and van der Schaar, M. 2018. INVASE: Instance-wise Variable Selection Using Neural Networks. In *Proceedings of the International Conference on Learning Representations*.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence Generative Adversarial Nets with Policy Gradient. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

Yu, M.; Chang, S.; Zhang, Y.; and Jaakkola, T. 2019. Rethinking Cooperative Rationalization: Introspective Extraction and Complement Control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4094–4103. Association for Computational Linguistics.

Zhang, J.; Bargal, S. A.; Lin, Z.; Brandt, J.; Shen, X.; and Sclaroff, S. 2018. Top-down Neural Attention by Excitation Backprop. *International Journal of Computer Vision* 126(10): 1084–1102.

Zhong, H.; Guo, Z.; Tu, C.; Xiao, C.; Liu, Z.; and Sun, M. 2018. Legal Judgment Prediction via Topological Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3540–3549. Association for Computational Linguistics.