Learning from the Best: Rationalizing Prediction by Adversarial Information Calibration

Lei Sha, Oana-Maria Camburu, Thomas Lukasiewicz

1Department of Computer Science, University of Oxford, UK 2Alan Turing Institute, London, 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

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
ing 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 \((x, y)\), \(x\) is the input with \(n\) features \(x = (x_1, x_2, \ldots, x_n)\), and \(y\) is the ground-truth corresponding label. The selector network \(Sel(\tilde{z}_{sym}|x)\) takes \(x\) as input and outputs \(p(\tilde{z}_{sym}|x)\), 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:

\[
u_i \sim U(0, 1), \quad g_i = -\log(-\log(u_i)) \quad (1)
\]

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

where \(U(0, 1)\) represents the uniform distribution between 0 and 1, and \(\tau\) is a temperature hyperparameter. Hence, we obtain the sampled mask \(m_i\) for each feature \(x_i\), and the vector symbolizing the rationale \(\tilde{z}_{sym} = (m_1x_1, \ldots, m_nx_n)\). Thus, \(\tilde{z}_{sym}\) is the sequence of discrete selected symbolic features forming the rationale.

**Predictor.** The predictor takes as input the rationale \(\tilde{z}_{sym}\) 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 \(\tilde{z}_{noro}^{1}\) then uses one feed-forward layer and a softmax layer to calculate the probability distribution over the possible predictions:

\[
\tilde{z}_{noro} = \text{Pred}(\tilde{z}_{sym}) \quad (3)
\]

\[
p(\hat{y}_{sp}|\tilde{z}_{sym}) = \text{Softmax}(W_{\hat{y}_{sp}}\tilde{z}_{noro} + b_{\hat{y}_{sp}}) \quad (4)
\]

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:

\[
L_{sp} = -\frac{1}{K} \sum_k \log p(\hat{y}_{sp}(k)|x^{(k)}) = -\frac{1}{K} \sum_k \log E_{\tilde{z}_{sym}|x^{(k)}} p(\hat{y}_{sp}(k)|\tilde{z}_{sym}) \quad (5)
\]

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 \(P_{sp}\), which receives the full original input \(x\) and transforms it into a dense feature vector \(\tilde{z}_{noro}\), 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

\[\text{Here, “noro” stands for neural feature (i.e., a neural vector representation) as opposed to a symbolic input feature.}\]
the input $x$, from which we sample a vector $z_{\text{nero}}$:

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

$$u \sim N((0,1), \quad z_{\text{nero}} = u\sigma + \mu$$

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

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 $L_{\text{guide}}$ is as follows:

$$L_{\text{guide}} = -\frac{1}{K} \sum_k \log p(y_{\text{guide}}^{(k)}|x^{(k)})$$

$$\leq -\frac{1}{K} \sum_k \mathbb{E}_{p(z_{\text{nero}}|x^{(k)})} \log p(y_{\text{guide}}^{(k)}|z_{\text{nero}}^{(k)}),$$

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 $I(x, \tilde{z}_{\text{sym}}) - I(z_{\text{sym}}, y)$ where the former term encourages the selection of few features, and the latter term encourages the selection of the necessary features. As $I(z_{\text{sym}}, y)$ is implemented by $L_{\text{sym}}$ (the proof is given in Appendix [A.1] in the extended paper), we only need to minimize the mutual information $I(x, \tilde{z}_{\text{sym}})$:

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

However, there is a time-consuming term $p(\tilde{z}_{\text{sym}}) = \sum_i p(\tilde{z}_{\text{sym}}|x)p(x)$, which needs to be calculated by a loop over all the instances $x$ 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)

$$I(x, \tilde{z}_{\text{sym}}) \leq \mathbb{E}_{x, \tilde{z}_{\text{sym}}} \left[\log \frac{p(\tilde{z}_{\text{sym}}|x)}{r_{\phi}(z)}\right].$$

Since $\tilde{z}_{\text{sym}}$ is a sequence of binary-selected features, we sum up the mutual information term of each element of $\tilde{z}_{\text{sym}}$ as the information bottleneck loss:

$$L_{ib} = \sum_i \sum_{\tilde{z}_i} p(\tilde{z}_i|x) \log \frac{p(\tilde{z}_i|x)}{r_{\phi}(\tilde{z}_i)}$$

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

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

$$L_{mi} = I(x, z_{\text{nero}}) \leq \mathbb{E}_{z_{\text{nero}}} \left[\log \frac{p(z_{\text{nero}}|x)}{p(z_{\text{nero}})}\right] = 0.5(\mu^2 + \sigma^2 - 1 - 2\log \sigma).$$

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 $z_{\text{nero}}$ to encode the information from the least-but-enough features, if we also require $z_{\text{nero}}$ and $z_{\text{nero}}$ 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 $\tilde{z}_{\text{nero}}$ or $z_{\text{nero}}$ 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 $z_{\text{nero}}$. The losses associated with the generator and discriminator are:

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

$$L_g = -\log D(z_{\text{nero}}).$$

2.3 Regularizing Rationales with Language Models

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

$$L_{im} = -\sum_i m_{i-1} \log p_m(m_1x_i|x_{<i}),$$

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.
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 $L_{\text{adv}}$. The discriminator is trained by $L_{d}$ from Eq. [13].

The adversarial-related losses are denoted by $L_{\text{adv}}$. The discriminator is trained by $L_{d}$ from Eq. [13].

$$L_{\text{adv}} = \lambda_{g}L_{g} + \lambda_{\text{guide}} + \lambda_{\text{sym}}L_{\text{sym}}$$ (16)

$$J_{\text{total}} = L_{sp} + \lambda_{ib}L_{ib} + L_{\text{adv}} + \lambda_{li}L_{li}$$ (17)

where $\lambda_{ib}$, $\lambda_{g}$, $\lambda_{\text{sym}}$, and $\lambda_{\text{sym}}$ are hyperparameters.

At training time, we optimize the generator loss $J_{\text{total}}$ and discriminator loss $L_{d}$ alternately until convergence. The detailed algorithm for training is given in Appendix C in the extended paper. At inference time, we run the selection-predictor model to obtain the prediction and the rationale $z_{\text{sym}}$.

### 3 Experiments

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

#### 3.1 Beer Reviews

**Data.** To provide a quantitative analysis for the extracted rationales, we use the BeerAdvocate 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.

**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 and HardKuma 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$-L_{\text{sym}}$.

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 baselines in terms of recall.
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.

| 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 | 56.7 | 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 | 31.3 | 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)| 97.7 | 71.7 | 82.8 | 13 | 94.8 | 42.3 | 58.5 | 7 | 89.4 | 56.9 | 69.5 | 7 |
| InfoCal−Ladv| 97.3 | 67.8 | 79.9 | 13 | 94.3 | 34.5 | 50.5 | 7 | 89.6 | 51.2 | 65.2 | 7 |
| InfoCal−Linh| 79.8 | 54.9 | 65.0 | 13 | 87.1 | 32.3 | 47.1 | 7 | 83.1 | 47.4 | 60.4 | 7 |

Table 2: One example of extracted rationales by different methods. Different colors correspond to different aspects: red: appearance, green: smell, and blue: palate.

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−Ladv, 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−Linh. This highlights the importance of semantical fluency of rationales, which are encouraged by 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 λβ. 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.

Table 2 shows an example of rationale extraction. Compared to the rationales extracted by Bernoulli and HardKuma, our method provides more fluent rationales for each aspect. For example, unimportant tokens like “and” (after “persistence”, in the Bernoulli method), and “with” (after “mouthful”, in the HardKuma method) were selected just because they are adjacent to important ones.

3.2 Legal Judgement Prediction

Datasets and Preprocessing. We use the CAIL2018 dataset[1] (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[2]. 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

---

1. [http://cail.oss-cn-qingdao.aliyuncs.com/CAIL2018_ALL_DATA.zip](http://cail.oss-cn-qingdao.aliyuncs.com/CAIL2018_ALL_DATA.zip)
2. [http://cail.cipsc.org.cn/index.html](http://cail.cipsc.org.cn/index.html)
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 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 3.

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−Ladv−Lib 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 λib), 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−Ladv−Lib 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 4. 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 4.

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
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 investigation transcripts and on-site photos; physical evidence identification documents; witnesses Liu A, Liu B, Testimony of Liu C, site inspection transcripts and on-site photos; physical evidence inspection reports and registration certificate, call list, description of the situation; identification transcripts; on-public prosecuion agency cited the following evidence: capture history, household with the victim Zou XX

Table 4: Human evaluation on the CAII2018 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.

|      | Law | Charges | ToP |
|------|-----|---------|-----|
|      | U   | C       | F   |
| Bernoulli | 4.37 | 3.96 | 3.45 |
| HardKuma   | 4.65 | 3.21 | 3.78 |
| InfoCal    | 4.72 | 3.78 | 4.02 |

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.

models can use different types of explanations, such as feature-based explanations (Lei, Barzilay, and Jaakkola 2016; Sha and Lukasiewicz 2021), information bottleneck methods (Tishby, Pereira, and Bialek 2000), and constrained generation (Shal 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 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 (Shal, 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 Kumaswamy 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.; Łukasiewicz, 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 Rationlization. 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.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. Neural Computation, 9(8): 1735–1780.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical Reparameterization with Gumbel-softmax. arXiv preprint arXiv:1611.10144.

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-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.

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

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed Representations of Words and
Appendices

A Proofs

A.1 Derivation of $I(\tilde{z}_{sym}, y)$

Theorem 2. Minimizing $-I(\tilde{z}_{sym}, y)$ is equivalent to minimizing $L_{sp}$.

Proof.

$$ I(\tilde{z}_{sym}, y) = \mathbb{E}_{\tilde{z}_{sym}, y} \left[ \frac{p(y|\tilde{z}_{sym})}{p(y)} \right] $$ \hspace{1cm} (18)

$$ = \mathbb{E}_{\tilde{z}_{sym}, y} \left[ \frac{p(y|\tilde{z}_{sym})}{p(y)} \right] - \mathbb{E}_{\tilde{z}_{sym}, y} p(y). $$

We omit $\mathbb{E}_{\tilde{z}_{sym}, y} p(y)$, because it is a constant, therefore, minimizing Eq. (18) is equivalent to minimizing the following term:

$$ \mathbb{E}_{\tilde{z}_{sym}, y} p(y|\tilde{z}_{sym}). $$ \hspace{1cm} (19)

As the training pair $(x, y)$ is sampled from the training data, and $\tilde{z}_{sym}$ is sampled from Sel$(\tilde{z}_{sym}|x)$, we have that

$$ \mathbb{E}_{\tilde{z}_{sym}, y} p(y|\tilde{z}_{sym}) = \mathbb{E}_x p(y|\tilde{z}_{sym}) p(\tilde{z}_{sym}|x) $$

$$ = \mathbb{E}_x p(y|x). $$ \hspace{1cm} (20)

We can give each $p(y|x)$ in Eq. (20) a $-\log$ to arrive to $-I(\tilde{z}_{sym}, y)$, then, it is not difficult to see that $-I(\tilde{z}_{sym}, y)$ has exactly the same form as $L_{sp}$. \hfill \square

A.2 Derivation of Equation (12)

$$ L_{mi} = I(x, z_{zero}) $$ \hspace{1cm} (21)

$$ = \mathbb{E}_{x, z_{zero}} \left[ \log \frac{p(z_{zero}|x)}{p(z_{zero})} \right] $$ \hspace{1cm} (22)

$$ = \mathbb{E}_{x, z_{zero}} p(x) \left[ \log \frac{p(z_{zero}|x)}{p(z_{zero})} \right] $$ \hspace{1cm} (23)

$$ \leq \mathbb{E}_{x, z_{zero}} \left[ \log \frac{p(z_{zero}|x)}{p(z_{zero})} \right] $$ \hspace{1cm} (24)

$$ = 0.5(\sigma^2 + \sigma^2 - 1 - 2 \log \sigma). $$ \hspace{1cm} (25)

A.3 Proof of Theorem 1

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

- $m_i' < \epsilon - i < m_i$, $(0 < \epsilon < 1)$, and
- $|p(m_i', x_i|x_i) - p(m_i, x_i|x_i)| < \epsilon$.

then the following two inequalities hold:

1. $L_{im}(\ldots, m_k, \ldots, m_n') < L_{im}(\ldots, m_k', \ldots, m_n)$.
2. $L_{im}(m_1, \ldots, m_k, \ldots, m_n') > L_{im}(m_1', \ldots, m_k, \ldots, m_n)$.

Proof. By Eq. (15) we have:

$$ L_{im}(\ldots, m_k, \ldots, m_n) = - \left[ \sum_{i \neq k} m_{i-1} \log P(m_i, x_i|x_{<i}) \right. $$

$$ + m_{k-1} \log P(m_k'x_k|x_{<k}) + m_k \log P(m_{k+1}x_{k+1}|x_{<k+1}) \bigg]. $$

Therefore, we have the following equation:

$$ L_{im}(\ldots, m_k', \ldots, m_n) - L_{im}(\ldots, m_k, \ldots, m_n') $$

$$ = - m_{k-1} \log p(m_k') + m_{k-1} \log p(m_k) $$

$$ - m_k \log p(m_{k+1}) + m_k \log p(m_{k+1}) $$

$$ - m_{n-1} \log p(m_{n-1}) + m_{n-1} \log p(m_{n-1}) $$

$$ - m_n \log p(m_n') + m_n \log p(m_n'), $$ \hspace{1cm} (27)

where, for simplicity, we use the abbreviation $p(m_k)$ to represent $p(m_k, x_k|x_{<k})$.

We also have that

$$ - m_{n-1} \log p(m_{n-1}) + m_{n-1} \log p(m_{n-1}) $$

$$ = (m_{n-1} - m_{n}) \log p(m_{n-1}) - m_{n-1} \log p(m_{n-1}) $$

$$ \geq \epsilon \log p(m_{k+1}) - \epsilon $$ \hspace{1cm} (30)

Since $p(m_k)$ are expected to have large probability values in the language model training process, we have that

$$ p(m_k) > \delta, $$

and, therefore, $-|\log \delta| < \log \frac{p(m_{k+1})}{p(m_k)} < |\log \delta|$. Hence, we have that:

$$ - m_{n-1} \log p(m_{n}) + m_k \log p(m_{k+1}) $$

$$ = (m_k - m_{n-1}) \log p(m_{k+1}) + m_{n-1} \log p(m_{k+1}) $$

$$ \geq \epsilon \log p(m_{k+1}) - |\log \delta| \geq (\epsilon - 1) |\log \delta|. $$ \hspace{1cm} (32)

Similarly, $-m_k \log p(m_{k+1}) + m_{k-1} \log p(m_k) \geq (1 - 2\epsilon) \log p(m_k) + m_k \log p(m_{k+1}) \geq (1 - 3\epsilon) |\log \delta|$. Therefore, the lower bound of the expression in Eq. (27) is:

$$ \inf = - (1 - \epsilon) \log p(m_k') + \epsilon \log p(m_{n+1}) $$

$$ + \epsilon \log p(m_{k+1}) - 2\epsilon |\log \delta| - \epsilon $$

$$ \geq - (1 - 3\epsilon) \log p(m_k') - 4\epsilon |\log \delta| - \epsilon > 0. $$ \hspace{1cm} (35)

This proves the statement of the theorem. \hfill \square

B Experimental Settings

B.1 Beer Reviews

The BeerAdvocate dataset. The training set of BeerAdvocate contains 220,000 beer reviews, with human ratings for each aspect. Each rating is on a scale of 0 to 5 stars, and it can be fractional (e.g., 4.5 stars). Lei, Barzilay, and Jaakkola (2016) have normalized the scores to represent $p(m_k, x_k|x_{<k})$. For each aspect, there are 80k–90k reviews for training and 10k reviews for validation.

http://people.csail.mit.edu/taolei/beer/
InfoCal with previous works on legal judgement prediction (Zhong 2018; Yang et al. 2019), we filter out these multi-label examples.

We also filter out instances where the charges and law articles occurred less than 100 times in the dataset (e.g., insulting the national flag and national emblem). For the term of penalty, we divide the terms into 11 non-overlapping intervals. These preprocessing steps are the same as in Zhong et al. (2018) and Yang et al. (2019), making it fair to compare our model with previous models.

We use Jieba2 for token segmentation, because this dataset is in Chinese. The word embedding size is set to 100 and is randomly initiated before training. The maximum sequence length is set to 1000. The architectures of the selector, predictor, and guider are all bidirectional LSTMs. The LSTM’s hidden size is set to 100. \( r_{\phi}(z_i) \) is the sampling rate for each token (0 for selected), which we set to \( r_{\phi}(z_i) = 0 = 0.9 \).

We search the hyperparameters in the following scopes:

\[
\lambda_{ib} \in [0.00, 0.10] \text{ with step 0.01, } \lambda_g \in [0.2, 2.0] \text{ with step 0.2, } \lambda_{mi} \in [0.0, 1.0] \text{ with step 0.1, and } \lambda_{lm} \in [0.00, 0.010] \text{ with step 0.001.}
\]

The best hyperparameters were found as follows: \( \lambda_{ib} = 0.0003, \lambda_g = 1, \lambda_{mi} = 0.1, \text{ and } \lambda_{lm} = 0.005. \)

We set \( r_{\phi}(z_i) \) to \( r_{\phi}(z_i = 0) = 0.999 \) and \( r_{\phi}(z_i = 1) = 0.001. \)

### B.2 Legal Judgement Prediction

The statistics of CAIL2018 dataset are shown in Table 6.

In the dataset, there are also many cases with multiple applicable law articles and multiple charges. To be consistent with previous works on legal judgement prediction (Zhong et al. 2018; Yang et al. 2019), we filter out these multi-label examples.

We also filter out instances where the charges and law articles occurred less than 100 times in the dataset (e.g., insulting the national flag and national emblem). For the term of penalty, we divide the terms into 11 non-overlapping intervals. These preprocessing steps are the same as in Zhong et al. (2018) and Yang et al. (2019), making it fair to compare our model with previous models.

We use Jieba for token segmentation, because this dataset is in Chinese. The word embedding size is set to 100 and is randomly initiated before training. The maximum sequence length is set to 1000. The architectures of the selector, predictor, and guider are all bidirectional LSTMs. The LSTM’s hidden size is set to 100. \( r_{\phi}(z_i) \) is the sampling rate for each token (0 for selected), which we set to \( r_{\phi}(z_i) = 0 = 0.9 \).

We search the hyperparameters in the following scopes:

\[
\lambda_{ib} \in [0.00, 0.10] \text{ with step 0.01, } \lambda_g \in [0.2, 2.0] \text{ with step 0.2, } \lambda_{mi} \in [0.0, 1.0] \text{ with step 0.1, and } \lambda_{lm} \in [0.00, 0.010] \text{ with step 0.001.}
\]

The best hyperparameters were found as follows: \( \lambda_{ib} = 0.0003, \lambda_g = 1, \lambda_{mi} = 0.1, \text{ and } \lambda_{lm} = 0.005. \)

We set \( r_{\phi}(z_i) \) to \( r_{\phi}(z_i = 0) = 0.999 \) and \( r_{\phi}(z_i = 1) = 0.001. \)

### C Language Model as Rationale Regularizer

Note that in Eq. 15, the target sequence of the language model \( P(m|x_i|x_{<i}) \) is formed of vectors instead of symbolic tokens, as in usually the case for language models. Therefore, we make some small changes in the pre-training of the language model. In typical language models, the last

---

Table 5: More instances from the BeerAdvocate dataset. In red the rationales for the appearance aspect, in green the rationales for the smell aspect, and in blue the rationales for the palate aspect.

|   | Gold | InfoCal |
|---|------|---------|
| 12oz bottle into my pint glass | looks decent, a brown color ( imagine that ! ) with a tan head | clear copper colored brew, medium cream colored head, floral hop background, grapefruit tones |
| 12oz bottle into my pint glass | looks decent, a brown color ( imagine that ! ) with a tan head | clear copper colored brew, medium cream colored head, floral hop background, grapefruit tones |
| 12oz bottle into my pint glass | looks decent, a brown color ( imagine that ! ) with a tan head | clear copper colored brew, medium cream colored head, floral hop background, grapefruit tones |
| 12oz bottle into my pint glass | looks decent, a brown color ( imagine that ! ) with a tan head | clear copper colored brew, medium cream colored head, floral hop background, grapefruit tones |
| 12oz bottle into my pint glass | looks decent, a brown color ( imagine that ! ) with a tan head | clear copper colored brew, medium cream colored head, floral hop background, grapefruit tones |

Table 6: Statistics of the CAIL2018 dataset.

|               | CAIL-small | CAIL-big |
|---------------|------------|----------|
| Cases         | 113,536    | 1,594,291|
| Law Articles  | 105        | 183      |
| Charges       | 122        | 202      |
| Term of Penalty | 11        | 11       |
layer is:

\[ p(x_i|x_{<i}) = \frac{\exp(h_i^T e_i)}{\sum_j \exp(h_i^T e_j)} \]  

(36)

where \( h_i \) is the hidden vector corresponding to \( x_i \) , \( e_i \) represents the output vector of \( x_i \), and \( V \) is the vocabulary. When we are modeling the language model in vector form, we only use a bilinear layer to directly calculate the probability in Eq. \( 36 \) :

\[ p(x_i|x_{<i}) = \sigma(h_i^T M e_i) \]  

(37)

where \( \sigma \) stands for sigmoid, and \( M \) is a trainable parameter matrix. To achieve this, we use negative sampling [Mikolov et al., 2013] in the training procedure.

The language model is pretrained using the following loss:

\[ L_{\text{pre}} = \sum_i \left[ \log \sigma(h_i^T M e_i) - E_{j \sim \text{p}(x_i)} \log \sigma(h_i^T M e_j) \right] \]  

(38)

where \( p(x_j) \) is the occurring probability (in the training dataset) of token \( x_j \).

D Training Procedure for InfoCal

The whole training process is illustrated in Algorithm 1.

E Human Evaluation Setup

Our annotators were asked the following questions, in order to assess the usefulness, completeness, and fluency of the rationales provided by our model.

E.1 Usefulness of Rationales

Q: Do you think the selected tokens/rationale are useful to explain the ground-truth label?

Please choose a score according to the following description. Note that the score is not necessarily an integer, you can give intermediate scores, such as 3.2 or 4.9 if you deem appropriate.

Figure 6: More instances from the CAIL2018 dataset. Left: the fact description (in Chinese). Right: the corresponding English translation of the fact description. In pink is the selected rationales.
Algorithm 1: Training process

Random initialization;
Pre-train language model by Eq. [38];

for each iteration \( i = 1, 2, \ldots \) do
  for each batch do
    Calculate the loss \( J_{\text{total}} \) for the
    sampler-predictor model and the guider
    model by Eq. [17];
    Calculate the loss \( L_D \) for the discriminator
    by Eq. [13];
    Update the parameters of selector-predictor
    model and the guider model;
    Update the parameters of the discriminator;
  end
end

• 5: Exactly. I can give the correct label only by seeing the
  given tokens.
• 4: Highly useful. Although most of the selected tokens
  lead to the correct label, there are still several tokens that
  have no relation to the correct label.
• 3: Half of them are useful. About half of the tokens can
  give some hint for the correct label, the rest are nonsense
  to the label.
• 2: Almost useless. Almost all of the tokens are useless,
  but there are still several tokens that are useful.
• 1: No Use. I feel very confused about the selected tokens,
  I don’t know which law article/charge/term of penalty the
  article belongs to.

E.2 Completeness of Rationales

Q: Do you think the selected tokens/rationale are enough to
explain the ground-truth label?

Please choose a score according to the following description. Note that the score is not necessary an integer, you can
give scores like 3.2 or 4.9, if you deem appropriate.
• 5: Exactly. I can give the correct label only by the given
  tokens.
• 4: Highly complete. There are still several tokens in the
  fact description that have a relation to the correct label,
  but they are not selected.
• 3: Half complete. There are still important tokens in the
  fact description, and they are in nearly the same number
  as the selected tokens.
• 2: Somewhat complete. The selected tokens are not
  enough. There are still many important tokens in the fact
  description not being selected.
• 1: Nonsense. All of the selected tokens are useless. None
  of the important tokens is selected.

E.3 Fluency

Q: How fluent do you think the selected rationale is? For
example: "He stole an iPhone in the room" is very fluent,
which should have a high score. "stole iPhone room" is just
separated tokens, which should have a low fluency score.

Please choose a score according to the following descrip-
tion. Note that the score is not necessary an integer, you can
give scores like 3.2 or 4.9, if you deem appropriate.
• 5: Very fluent.
• 4: Highly fluent.
• 3: Partial fluent.
• 2: Very unfluent.
• 1: Nonsense.

F More Examples of Rationales

F.1 BeerAdvocate

We list more examples of rationales extracted by our model
for the BeerAdvocate dataset in Table 5.

F.2 Legal Judgement Prediction

More examples of rationales extracted by our model for the
legal judgement tasks are shown in Table 6.