Distribution Matching for Rationalization

Yongfeng Huang[2*], Yujun Chen[1], Yulun Du[1], Zhilin Yang[1]
[1]Recurrent AI, Beijing
[2]Tsinghua University, Beijing
huangyf17@tsinghua.org.cn, chenyujun@rcrai.com, duyulun@rcrai.com, kimi_yang@rcrai.com

Abstract

The task of rationalization aims to extract pieces of input text as rationales to justify neural network predictions on text classification tasks. By definition, rationales represent key text pieces used for prediction and thus should have similar classification feature distribution compared to the original input text. However, previous methods mainly focused on maximizing the mutual information between rationales and labels while neglecting the relationship between rationales and input text. To address this issue, we propose a novel rationalization method that matches the distributions of rationales and input text in both the feature space and output space. Empirically, the proposed distribution matching approach consistently outperforms previous methods by a large margin. Our data and code are available[1].

Introduction

In many real-world NLP applications, interpretability is an important objective for model development because it is crucial for human users to understand, verify, and trust the machine predictions. Among other possibilities, rationalization is a learning paradigm that extracts key text pieces as rationales to justify and interpret model predictions (Lei, Barzilay, and Jaakkola 2016). Specifically, Lei, Barzilay, and Jaakkola (2016) uses a generator to selectively extract rationales from the original input, and a classifier is applied on the rationales to predict the classification labels. This can be viewed as a cooperative game between the generator and the classifier to maximize the mutual information between the rationales and the labels (Chen et al. 2018). In other words, this is based on the desideratum that the extracted rationales are predictive of the classification labels. Different variants of rationalization methods have been proposed under this framework, which additionally consider other desiderata such as the dependency between labels, rationales, and the complement of rationales (Chang et al. 2019, 2020; Yu et al. 2019).

In this work, we argue that it is crucial to incorporate the following desideratum into modeling—the rationales and the original full input text should have similar feature and output distributions when the same classifier is applied. By definition, rationales represent key text pieces that are actually used for predicting the labels. The definition has a two-folded implication. First, the rationales should have a similar feature distribution to the input text because intermediate feature representations directly reflect how the model processes natural language. Second, since ultimately the probability outputs are used for classification, the rationales should have a similar output distribution to the input text. For example, consider a review "this is a great movie". A well-trained sentiment classifier mainly uses the rationale "great movie" for prediction. When the classifier is applied, "great movie" and "this is a great movie" should have similar feature and output distributions.

However, the aforementioned prior has not been effectively leveraged in previous approaches. As a solution, we propose a novel distribution matching approach for rationalization. In the feature space, we impose a regularization term that minimizes the central moment discrepancy (CMD) (Zellinger et al. 2017) between the full input features and the rationale features. In the output space, a teacher-student distillation loss (Hinton, Vinyals, and Dean 2015) is employed to minimize the cross entropy loss between the full input predictions and the rationale predictions. Our approach is a plug-and-play improvement that is applicable to different rationalization variants.

We evaluate the proposed distribution matching approach on widely-used rationalization benchmarks—the beer review dataset (McAuley, Leskovec, and Jurafsky 2012) and hotel review dataset (Bao et al. 2018). We use the game-theoretic class-dependent model (Chang et al. 2019) as our base model. Empirical results show that distribution matching substantially improves over the baseline and outperforms all considered previous methods. It is also observed that both feature space matching and output space matching contribute to the overall performance.

To summarize, our work makes the following four contributions:

- We analyze the rationalization framework and uncover the issue that existing approaches neglect the relationship between rationales and input text.
- We propose to impose an inductive bias that rationales

[*]Work done during an internship at Recurrent AI

[1]https://github.com/kochsnow/distribution-matching-rationality

should have similar feature and output distributions with input text so as to improve the faithfulness of rationalization.

- We achieved state-of-the-art results with substantial gains on multiple settings.

Related Work

Interpretability

There are multiple lines of research in the area of learning interpretable models for NLP tasks. Roughly, there are three categories—post-hoc explanation methods, extractive rationalization methods, and the self-explaining model-based approach.

Extractive Rationalization. Extractive rationalization selects pieces of text from the input to form rationales that justify the prediction. Multiple variants were proposed to improve over the original framework (Lei, Barzilay, and Jaakkola 2016). Chang et al. (2019) introduced a game-theoretic framework where the rationale generator is dependent on the class labels. Yu et al. (2019) employed a similar idea and additionally employed constraints on the complement of rationales. Other approaches were based on the information bottleneck (Paranjape et al. 2020), latent variable models (Bastings, Aziz, and Titov 2019), and learning environment-invariant representations (Chang et al. 2020). However, none of these previous methods consider the relationship between rationales and the full input in terms of feature distribution.

Recently, a new benchmark (DeYoung et al. 2019) was introduced with labeled rationales available for training. This enables directly finetuning pretrained models (Peters et al. 2018; Radford 2018; Devlin et al. 2019; Yang et al. 2019) to predict the rationales in an end-to-end fashion. However, in this work, we focus on the conventional unsupervised learning setting because rationale groundtruth is not available for most real-world tasks.

Post-Hoc Explanation. The post-hoc methods do not require specific additional efforts during training and explainations are computed after training is finished. Most of these approaches invetsigate the gradient saliency in a trained neural network. For example, Sundararajan, Taly, and Yan (2017); Smilkov et al. (2017); Bao et al. (2018) studied the integrated gradients of a model for interpretability.

Model-Based Approach. Another line of research focuses on developing models that self-explain the results. For example, module networks (Andreas et al. 2016) learn structures in addition to weights so that the learned structure can be used to interpret how the model processes and reasons over natural language. Johnson et al. (2017) adapted the concept of module networks to the vision domain.

Knowledge Distillation

Hinton, Vinyals, and Dean (2015) proposed the teacher-student framework of knowledge distillation that transfers knowledge from a teacher model to a student model by optimizing the cross entropy loss. The most typical use case of knowledge distillation is model compression. A small model is distilled from a large pretrained model to achieve a more desirable complexity-effectiveness trade off (Sanh et al. 2019; Jiao et al. 2019). Knowledge distillation is also known to improve performance when the student model is of comparable size with the teacher model (Yim et al. 2017; Furlanello et al. 2018; Wang and Yoon 2020) because it provides soft, continuous labels for more effective training. (Yoon, Jordon, and van der Schaar 2019) used knowledge distillation in selecting instance-wise features which is similar with rationalization.

Learning Domain-Invariant Representations

There are two main categories for learning domain-invariant representations—adversarial training and distribution matching. Adversarial training introduces an adversarial game where a discriminator learns to distinguish features extracted by an encoder (Ganin and Lempitsky 2015). Distribution matching, on the other hand, is based on minimizing the distance between distributions in various forms (Zellinger et al. 2017; Gretton et al. 2006; Li, Swersky, and Zemel 2015).

Distribution Matching for Rationalization

In this section, we first introduce the standard rationalization framework (Lei, Barzilay, and Jaakkola 2016) and discuss our proposed method. Then we consider a more advanced rationalization variant (i.e., the game-theoretic approach introduced by Chang et al. (2019)) and discuss how to implement our framework on it.

Preliminaries: The Rationalization Framework

The input to the rationalization framework (Lei, Barzilay, and Jaakkola 2016) is a text sequence $\mathbf{x} = (x_1, x_2, \cdots, , x_l)$ of length l, where each $x_i \in \mathcal{V}$ denotes the i-th token and $\mathcal{V}$ is the vocabulary. A generator g is applied on $\mathbf{x}$ to obtain the rationale mask $\mathbf{z}$, i.e.,

$$\mathbf{z} = g(\mathbf{x})$$

where the rationale mask $\mathbf{z}$ is represented as a sequence of binary variables $\mathbf{z} = (z_1, z_2, \cdots, z_l)$. Each entry $z_i = 1$ means x_i is selected as part of the rationales and $z_i = 0$ denotes the opposite. In other words, given the input text $\mathbf{x}$ and the mask $\mathbf{z}$, the rationale can be obtained as $\{x_i | z_i = 1\}$.

A classifier c is applied on top of the rationale to obtain the model output distribution $\hat{p}(Y)$. Computationally, we use a lookup table to obtain the input embeddings $\mathbf{e}(\mathbf{x})$ and feed the masked embeddings to the classifier c as follows:

$$\hat{p}(Y) = c(\mathbf{z} \odot \mathbf{e}(\mathbf{x}))$$

where $\odot$ denotes element-wise multiplication and $\hat{p}(Y)$ is a probability distribution over the classes.

Let y be the groundtruth label in the label space $\mathcal{Y}$. The classification loss is written as a standard cross entropy loss:

$$l_{\text{cls}} = -\log \hat{p}(Y = y)$$

Additionally, it is desirable to control the sparsity and compactness of the rationales. To achieve this goal, the following regularization is applied:

$$\Omega(\mathbf{z}) = \lambda_1 \|\mathbf{z}\|_1 + \lambda_2 \sum_{i=2}^{l} |z_i - z_{i-1}|$$

with λ_1 and λ_2 being the coefficients.

The generator and the discriminator are jointly trained to minimize the overall loss function

$$\min_{g,c} l_{\text{cls}} + \Omega(\mathbf{z}).$$

Since $\mathbf{z}$ is discrete, the loss function is not differentiable w.r.t. the generator parameters. Methods like straight-through (Bengio, Léonard, and Courville 2013) can be used for optimization.

Distribution Matching for Rationalization (DMR)

Now we introduce our DMR method to improve rationalization based on distribution matching. Figure 1 illustrates the DMR model in comparison with the baseline RNP (Lei, Barzilay, and Jaakkola 2016).

The underlying assumption and desideratum of our distribution matching approach is that the rationales and the original full input text should have similar feature and output distributions when the classifier is applied. Intuitively, interpreting model predictions is to explain how the classifier processes input information. Since rationales are to interpret and justify the predictions, when the input contains only the rationales, the classifier should learn features and make predictions in a very similar way compared to using the full original text as input. Based on this intuition, we propose to encourage similar distributions in both the feature and output spaces between the rationales and the full text input.

Feature Space Matching The classifier c can be instantiated as different models such recurrent neural networks (Hochreiter and Schmidhuber 1997), convolutional networks (Waibel et al. 1989) and Transformers (Vaswani et al. 2017). Let f be the function in classifier c that maps word embeddings to network output features—e.g., the output of a max-pooling layer. Given a text sample $\mathbf{x}_i$ with rationale $\mathbf{z}_i$, it follows that $f(\mathbf{e}(\mathbf{x}_i))$ and $f(\mathbf{z}_i \odot \mathbf{e}(\mathbf{x}_i))$ are the features of the full input text and the rationales respectively. Denote these two features as $\mathbf{f}_i^x$ and $\mathbf{f}_i^z$ respectively.

Given a batch of N training samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ with rationales $\{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N\}$ computed by the generator g, we employ the classifier c to compute features $\{\mathbf{f}_1^x, \mathbf{f}_2^x, \cdots, \mathbf{f}_N^x\}$ and $\{\mathbf{f}_1^z, \mathbf{f}_2^z, \cdots, \mathbf{f}_N^z\}$. A central moment discrepancy regularizer (Zellinger et al. 2017) is employed to match the distributions in the feature space:

$$l_{\text{fm}} = \|\mathbf{E}_x - \mathbf{E}_z\|_2 + \sum_{k=2}^{K} \|\mathbf{C}_k^x - \mathbf{C}_k^z\|_2$$

with

$$\mathbf{E}_x = \frac{1}{N} \sum_{i=1}^{N} \mathbf{f}_i^x$$

$$\mathbf{C}_k^x = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{f}_i^x - \mathbf{E}_x)^k$$

where $\mathbf{E}_x$ the is empirical expectation of the features, and $\mathbf{C}_k^x$ is k-th order central moments of the feature coordinates. $\mathbf{E}_z$ and $\mathbf{C}_k^z$ are defined in a similar way. In practice, we compute the central moments up to the fifth order, i.e., $K = 5$. It is assumed that the features are distributed in the interval $[0, 1]$—e.g., the output of the sigmoid function; in other cases, constants might be added before each term (Zellinger et al. 2017).

The feature space matching loss l_{fm} enforces the rationales and the full input to have similar feature distributions when the classifier c is applied.

Output Space Matching A straightforward method to add training signals in the output space is to use a normal classification loss as in (Lei, Barzilay, and Jaakkola 2016). However, following (Hinton, Vinyals, and Dean 2015), we believe only a categorical label is not sufficient to provide useful training signals. Our goal is to ensure that the rationales and the full input have similar output distributions, so knowledge distillation (Hinton, Vinyals, and Dean 2015) is used for distribution matching in the probability output space.

Specifically, we first pretrain a teacher classifier c_t that maps the full input text to the probability space using a standard classification loss. Let $\hat{p}_t(Y) = c_t(\mathbf{e}(\mathbf{x}))$ be the teacher model distribution of sample $\mathbf{x}$. The output space matching loss is written as the cross entropy between the teacher distribution $\hat{p}_t(Y)$ and the student distribution $\hat{p}(Y)$:

$$l_{\text{om}} = \sum_{y=1}^{|\mathcal{Y}|} -\hat{p}_t(Y = y) \log \hat{p}(Y = y).$$

Overall Loss Function The overall loss function is formulated as the weighted sum of the feature and output space matching losses, along with the normal rationalization losses, i.e.,

$$\min_{c} l_{\text{cls}} + \lambda_3 l_{\text{fm}} + \lambda_4 l_{\text{om}}$$

$$\min_{g} l_{\text{cls}} + \Omega(\mathbf{z})$$

where λ_3 and λ_4 are the coefficients of the loss terms. Note that we apply the matching losses only on the classifier and do not backpropagate the gradients of l_{fm} and l_{om} to the generator g. Also the regularizer $\Omega(\mathbf{z})$ only depends on the generator. Although gradients from discriminator d are not directly passed to g, the generators essentially benefit from the losses. Since we measure the accuracy of rationale prediction, the results improve if and and only if the generators improve. Therefore, the generators benefit a lot from our two regularization terms (though in an indirect manner).

Extensions and Implementation

Our above derivation is based on the original rationalization framework (Lei, Barzilay, and Jaakkola 2016). However, our approach is general and applicable to different backbone methods. In our preliminary study, we experimented with multiple variants that improve over the original method and

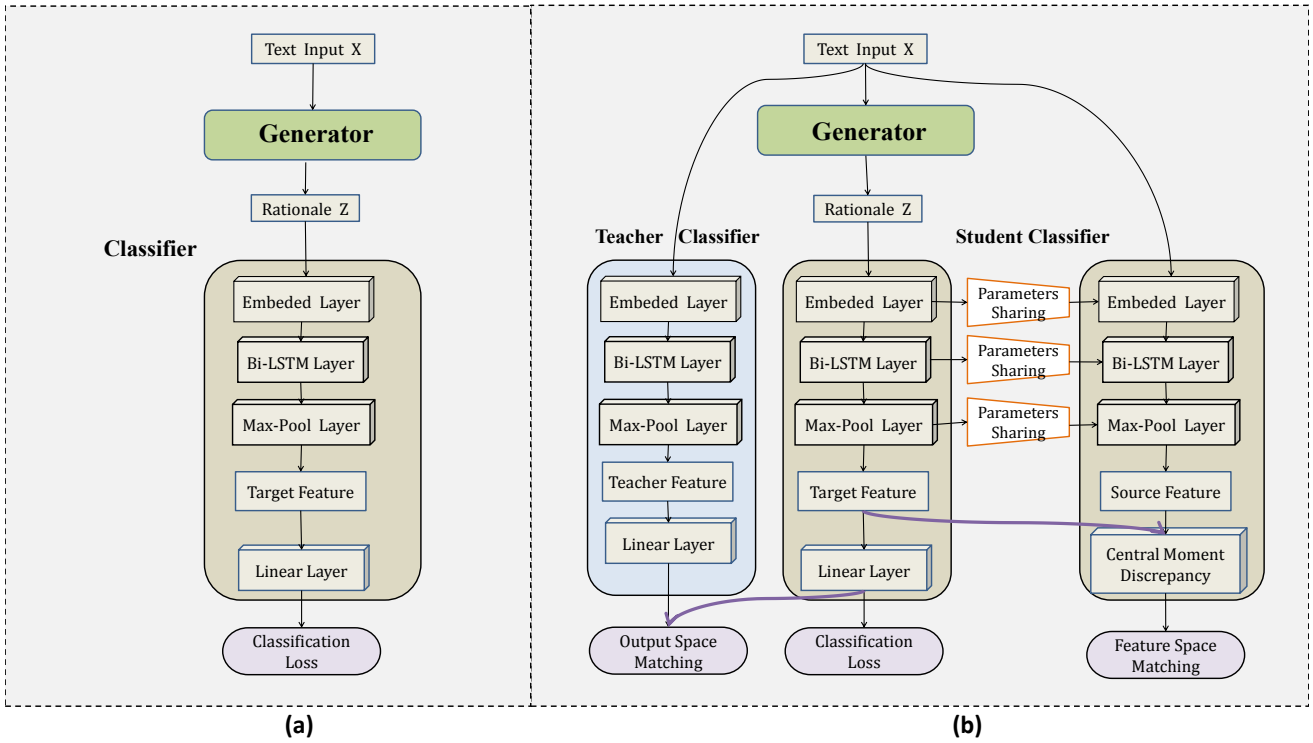

Figure 1: Comparison of (a) the baseline RNP framework, and (b) our proposed DMR framework. We run the classifier on the full input text for feature matching, and also train a student classifier for output matching.

found that the CAR framework (Chang et al. 2019) works particularly well. The CAR framework feed ground-truth label to the generator and proposed the cooperative and adversarial game mechanism.

We adopt a two-stage training scheme in our implementation. In the first stage, the teacher classifier c_t is pretrained on the full dataset. After pretraining the teacher classifier c_t, we jointly train the student classifiers and the generators using the aforementioned losses.

Experiments

Datasets

To evaluate the performance of our DMR framework, we use the multi-aspect beer and hotel datasets, which are commonly used in the field of rationalization.

Beer reviews: The beer review dataset (McAuley, Leskovec, and Jurafsky 2012) is a multi-aspect sentiment classification dataset, where each review of a beer consists of a plain-text comment and ratings from three aspects including appearance, aroma and palate.

Hotel reviews: The hotel review dataset (Bao et al. 2018) is another multi-aspect sentiment classification dataset. The dataset contains reviews of hotels from three aspects including location, cleanliness, and service. Each review also has a rating on a scale of 0-5 stars.

We preprocess both datasets in the same setting as (Chang et al. 2019) for fair comparison.

Baselines

We consider the following baselines for comparison in our experiments:

- **RNP:** RNP is the original rationalization framework proposed in (Lei, Barzilay, and Jaakkola 2016). The generator selects text segments as rationales, and the predictor is fed with rationales for label classification. RNP aims to maximize the mutual information between rationales and labels and the rationales are constrained to be both sparse and continuous.
- **3PLAYER:** The 3PLAYER method is an enhancement of RNP (Yu et al. 2019), which alleviates the degeneration problem of RNP by introducing an extra complement predictor. The complement predictor tries to maximize the predictive accuracy from unselected words and plays an adversarial game with generator.
- **INVRAT:** INVRAT introduces a game-theoretic invariant criterion as the objective and aims to learn environment invariant representations (Chang et al. 2020).
- **CAR:** CAR proposes a game theoretic approach to class-wise selective rationalization (Chang et al. 2019). The approach produces both positive and negative rationales.

Beer	Appearance				Aroma				Palate			
	S	P	R	F	S	P	R	F	S	P	R	F
CAR	11.9	76.2	49.3	59.9	10.3	50.3	33.3	40.1	10.2	**56.6**	46.2	50.9
DMR (ours)	11.7	**83.6**	**52.8**	**64.7**	11.7	**63.1**	**47.6**	**54.3**	10.7	55.8	**48.1**	**51.7**
Hotel	Location				Service				Cleanliness			
	S	P	R	F	S	P	R	F	S	P	R	F
CAR	10.6	46.6	58.1	51.7	11.7	40.7	41.4	41.1	10.3	29.0	33.8	31.2
DMR (ours)	10.7	**47.5**	**60.1**	**53.1**	11.6	**43.0**	**43.6**	**43.3**	10.3	**31.4**	**36.4**	**33.7**

Table 1: Comparison with *CAR* on both the beer review dataset and the hotel review dataset. S, P, R, and F1 represent the sparsity level, precision, recall, and F1 score respectively. We use the same (or similar) sparsity levels as previous work for fair comparison. All the baseline results are taken from (Chang et al. 2019).

Beer	Appearance				Aroma				Palate			
	S	P	R	F	S	P	R	F	S	P	R	F
RNP	7.9	13.5	5.8	8.1	8.4	30.3	15.3	20.3	9.1	28.2	17.2	21.4
3PLAYER	7.9	15.8	6.8	9.5	8.4	48.9	24.4	32.6	9.1	14.2	8.5	10.7
INVRAT	7.9	49.5	20.9	29.3	8.4	48.2	24.4	32.4	9.1	32.8	20.0	24.9
DMR (ours)	7.9	**80.1**	**34.7**	**48.6**	8.9	**50.3**	**28.9**	**36.7**	9.6	**49.7**	**38.2**	**43.2**
RNP	15.8	13.5	11.3	12.3	16.8	34.3	34.2	34.3	18.1	19.8	23.8	21.6
3PLAYER	15.8	15.6	13.5	14.5	16.8	35.7	35.9	35.8	18.1	20.7	24.9	22.6
INVRAT	15.8	58.0	49.6	53.5	16.8	42.7	42.5	42.6	18.1	**44.0**	**52.8**	**48.0**
DMR (ours)	15.7	**61.5**	**52.0**	**56.4**	16.8	**47.6**	**51.3**	**49.4**	15.7	39.4	49.7	44.0
RNP	23.7	26.3	33.1	29.3	25.2	40.0	60.1	48.0	27.2	19.2	33.8	24.5
3PLAYER	23.7	12.6	15.9	14.0	25.2	33.0	49.7	39.7	27.2	22.0	39.3	28.2
INVRAT	23.7	54.0	**69.2**	60.7	25.2	44.7	67.4	53.8	27.2	26.5	46.9	33.9
DMR (ours)	21.2	**58.9**	67.4	**62.9**	25.0	**44.7**	**71.8**	**55.1**	27.2	**28.0**	**61.3**	**38.4**

Table 2: Comparsion with *RNP*, *3PLAYER* and *INVRAT* on the beer review dataset. S, P, R, and F1 represent the sparsity level, precision, recall, and F1 score respectively. We use the same (or similar) sparsity levels as previous work for fair comparison. All the baseline results are taken from (Chang et al. 2020).

	appearance			aroma			palate		
input	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
rationales	92.09	88.56	90.29	93.47	79.23	85.77	93.87	71.79	81.36
full texts	91.54	91.88	91.71	94.34	79.26	86.15	94.52	70.08	80.49

Table 3: Comparison of classification performances using rationales and full texts

This is the direct baseline of our results because we use CAR as our backbone model.

To seek for fair comparisons, the generators, the predictors, and the discriminators of all the baselines and our method use the same architecture as in the previous work (Chang et al. 2019). In addition, the sparsity and continuity constraints follow the same form and all methods are adjusted to have a comparable level of sparsity in the experiments. In our experiment, we implement our DMR framework directly based on the CAR model.

Following previous work (Chang et al. 2019), the hidden unit size and the embedding dimension of the teacher discriminator are set as 100, while those of the generators and the student discriminator are set as 102 for two extra class label dimensions.

Quantitative Evaluation

In this section, we evaluate the performance of DMR and compare its performance against with the state of arts methods on the beer and the hotel review datasets.

We train our models using a balanced training dataset as in (Chang et al. 2019) and evaluate the performances on the test sets with human annotated rationales. Since rationales generated from CAR were based on ground truth label, we also infer our rationales condition on true labels. As results shown in Table 1, DMR outperforms CAR in all aspects of two datasets.

To further compare with *RNP*, *INVRAT* and *3PLAYER*, we adjust the sparsity levels to obtain the same rationale lengths on the beer dataset as in (Chang et al. 2020). The data split and processing of the beer review dataset in (Chang et al. 2020) are not available. For fair comparsion, we train and validate the models using the same beer review dataset but with different data split and processing, and the performances of rationales are evaluated on the same annotated test set. As shown in Table 2, DMR obtains the best performance in almost all metrics and sparsity combinations. In addition, our DMR method does not need ground-truth labels to generate rationales, as the predicted labels provided

Beer	Appearance				Aroma				Palate			
	S	P	R	F	S	P	R	F	S	P	R	F
DMR	8.8	**78.4**	**37.3**	**50.6**	8.8	48.7	**27.6**	**35.3**	8.9	**60.2**	**43.2**	**50.3**
- fm	8.9	74.5	35.7	48.3	8.4	49.5	26.7	34.7	8.9	58.6	42.2	49.1
- fm&om	8.8	70.0	33.4	45.2	7.7	**50.2**	24.8	33.2	8.9	54.3	38.9	45.3
DMR	12.3	**82.6**	**55.0**	**66.1**	13.0	**61.8**	**51.5**	**56.1**	11.5	**55.8**	**51.6**	**53.6**
- fm	12.4	77.0	51.5	63.5	12.0	60.4	46.5	52.6	12.3	51.3	50.5	50.9
- fm&om	12.3	74.5	49.6	59.6	12.9	55.2	45.6	50.0	12.1	50.6	49.1	49.8
DMR	16.0	**72.8**	**63.0**	**67.5**	16.2	52.3	**54.6**	**53.4**	16.4	**45.2**	**59.5**	**51.4**
- fm	16.0	69.0	59.7	64.0	15.5	**52.4**	52.2	52.3	16.0	38.4	49.4	43.2
- fm&om	16.0	63.9	55.4	59.4	16.1	36.2	37.5	36.8	16.3	37.7	49.3	42.7
DMR	24.0	**57.7**	**74.7**	**65.1**	24.1	**46.1**	**71.3**	**56.0**	23.8	**36.6**	**70.2**	**48.1**
- fm	23.9	54.1	69.9	61.0	24.0	44.0	67.8	53.4	24.0	29.8	57.4	39.2
- fm&om	23.9	53.9	69.7	60.8	24.1	43.0	66.6	52.2	23.9	28.5	54.8	37.5

Table 4: Ablation study. "- fm" means removing the feature space matching loss, and "- fm&om" means removing both feature and output space matching losses.

Beer	Appearance				Aroma				Palate			
	S	P	R	F	S	P	R	F	S	P	R	F
DMR_{CORAL}	11.9	79.0	50.8	61.9	11.7	60.3	45.2	51.6	10.1	47.0	38.0	42.0
DMR_{MMD}	11.7	81.0	51.4	62.9	11.5	50.0	36.9	42.4	10.6	44.5	38.0	41.0
DMR (ours)	11.7	**83.6**	**52.8**	**64.7**	11.7	**63.1**	**47.6**	**54.3**	10.7	**55.8**	**48.1**	**51.7**

Table 5: Comparison of Different Matching Loss Selected.

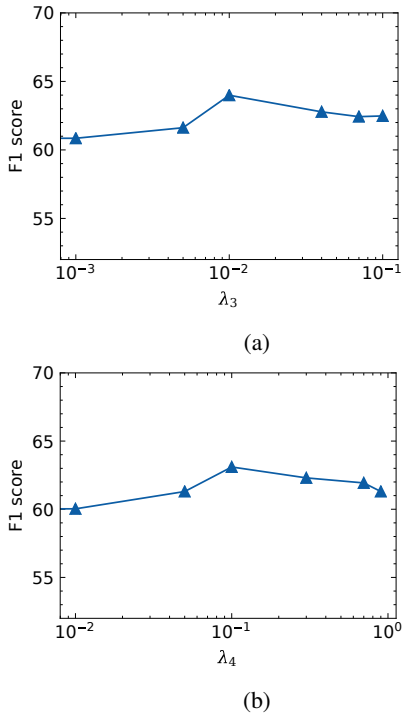

Figure 2: Parameter Sensitivity of λ_3 and λ_4

by the teacher classifier are used instead.

The classification results in the Table 3 show that better rationalization does not substantially improve classification, but with our method, training a classifier on the rationales is able to achieve performance comparable to using full text.

The results in Table 1 and 2 reveal the effectiveness of distribution matching. Compared with existing approaches, DMR is able to extract more accurate rationales and the advantages can be extended to all sparsity levels. And Table 3 presents that extracted rationales by our DMR are comparable to full texts on the classification.

Ablation Study

Effectiveness of our Matching Losses We conducted ablation studies to understand the importance of feature space matching and output space matching in the training process.

In Table 4, we show the performances of DMR with different matching losses removed (row begin with "-") under different levels of sparsity. The rows with "-fm" stand for models that have the feature space matching loss removed, and the rows with "-fm&om" correspond to models trained without neither the feature space matching nor the output space matching.

As shown in the table, both feature space matching and output space matching contribute to the performance of our method. The improvements brought by both methods are consistent and substantial across different settings, which validates our motivation in the previous sections.

Comparison of different Feature Space Matching Losses Many studies have shown that CMD outperforms MMD (Li, Swersky, and Zemel 2015) and CORAL (Sun and Saenko 2016) for its efficiency and invariance to different styles

Aspect	DMR Rationale	CAR Rationale
Appearance	**tangerine pour with a small white head that clings to the edge of the glass .** the hopping is smooth and mild , but the bitterness does gradually build , although it reminded me more of an english bitter instead of an american ipa . the malts come out as fruity with some honey . medium-light body with decent carbonation . i ca n't give it a glowing review because its not a great beer . pyramid seems to be very hit and miss , and this is a miss .	**tangerine pour with a** small white head that clings to the edge of the glass . the hopping is smooth and mild , but the bitterness does gradually build , although it reminded me more of an english bitter instead of an american **ipa . the** malts come out as fruity with some honey . medium-light body with decent carbonation . i ca n't give it a glowing review because its not **a great beer . pyramid seems** to be very hit and miss , and this is a miss .
Aroma	appearence: pours a crystal clear amber with a thin , bubbly white head that dies to a collar. **smell : solid belgian pale ale malt and hop characteristics throughout, with that perfect yeast tinge. taste** and mouthfeel : rich , full , thirst-quenching , and smooth . very balanced and tasty , with the perfect mouthfeel .	appearence : pours a crystal clear amber with a thin , bubbly white head that dies to a collar . smell : solid belgian pale ale malt and hop characteristics throughout , with that perfect yeasttinge. taste **and mouthfeel : rich , full , thirst-quenching , and smooth . very balanced and tasty , with the** perfect mouthfeel .
Palate	pours an amber with an orange hue . two inch white head fades quickly . very little lacing . smells like bread , not much else . taste some sweet malt , and grass . not much better than a macro . **lighter body with lots of carbonation. not** a lot of flavor but this is a refreshing beer . i have no problem drinking these , i just would n't pursue it .	**pours an** amber with an orange hue . two inch white head fades quickly . very little lacing . smells like bread , not much else . taste some sweet malt , and grass . not much better than a macro . **lighter body with lots of carbonation.** not a lot of flavor but **this is a** beer. i have no problem drinking these , i just would n't pursue it .

Table 6: Examples of rationales generated by our DMR method and the baseline CAR method on the three aspects of the beer dataset. Underlined words are the human annotated labels, and bold word are predicted positive rationales.

Aspect	DMR Rationale
Appearance	**the beige head is comprised of** medium-sized bubbles and **slowly , but inevitably ,** recedes to a thin strip ; there is some lacing adhering to the glass . black malt (i think) lends the beer a not pleasantly bitter and toasted flavour .
Aroma	pours with a nice foamy frothy off white head that lasts and a little lace . color is an ever so slightly **hazy amber . aroma is malty , grassy , hoppy , and bready beer . flavor 's very similar along with** pretzels and with bitterness coming out at the end.
Palate	typical of a hefeweizen taste shows notes of orange and the typical hefeweizen taste (cloves and bananas) . **smooth and very effervescent . almost no** bitterness too . **very drinkable and refreshing .** a nice **hefeweizen** .

Table 7: Failure examples of rationales generated by our DMR method on the beer dataset. The underlined words are the human annotated labels, and the bold words are predicted positive rationales.

of input. In our experiments, we also find that using CMD matching can provide more stable and generally better results. Results in Table 5 show that using CMD can always lead to the best results with respect to the F1 scores on all aspects of beer review dataset.

Sensitivities of Hyper-parameters We also studies the influences of different values of hyper-parameters λ_3 and λ_4 in section , respectively. As presented in Figure 2, the overall performance of our method is not sensitive to either the values of λ_3 or λ_4.

Case Studies

In this section, we visualize the rationales generated by our DMR framework and the CAR framework. As presented in Table 6, rationales generated by DMR are more accurate. For example, the DMR framework selects exactly the same rationales as the human annotations, while the CAR framework only finds a few related words combined with many unrelated words, specially in appearance and aroma aspects, which shows that DMR can extract meaningful and accurate rationales.In addition, we also show some failure examples in Table 7.

Conclusions

In this paper, we propose a novel rationalization framework based on distribution matching called DMR. DMR aims to match rationales and the input text in both the feature space and the output space. For feature space matching, we formulate it as minimization of the central moment discrepancy (CMD) between input text features and the rationale features. For the output space matching, we transfer the knowledge from the output distribution of the original full text to that of the rationales in a teacher-student distillation framework. The framework is highly flexible and can be applied to many existing rationale extraction methods. Extensive experiments show that the DMR framework outperforms state-of-the-art methods in most experimental settings. Ablation studies show that both feature space matching and output space matching contribute to the final performance. Moreover, case analysis show that DMR provides more meaningful and accurate rationales. In the future, it will be intriguing to apply our method to more interpretability settings such as non-classification tasks.

Acknowledgments

This work is funded by National Key R&D Program of China (2020AAA0105200) and supported by Beijing

Academy of Artificial Intelligence (BAAI).

References

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Learning to Compose Neural Networks for Question Answering. *ArXiv* abs/1601.01705.

Bao, Y.; Chang, S.; Yu, M.; and Barzilay, R. 2018. Deriving Machine Attention from Human Rationales. In *EMNLP*.

Bastings, J.; Aziz, W.; and Titov, I. 2019. Interpretable Neural Predictions with Differentiable Binary Variables. In *ACL*.

Bengio, Y.; Léonard, N.; and Courville, A. C. 2013. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *ArXiv* abs/1308.3432.

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

Chang, S.; Zhang, Y.; Yu, M.; and Jaakkola, T. S. 2020. Invariant rationalization. *arXiv preprint arXiv:2003.09772* .

Chen, J.; Song, L.; Wainwright, M. J.; and Jordan, M. I. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In *ICML*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv* abs/1810.04805.

DeYoung, J.; Jain, S.; Rajani, N. F.; Lehman, E.; Xiong, C.; Socher, R.; and Wallace, B. C. 2019. ERASER: A Benchmark to Evaluate Rationalized NLP Models. *arXiv preprint arXiv:1911.03429* .

Furlanello, T.; Lipton, Z. C.; Tschannen, M.; Itti, L.; and Anandkumar, A. 2018. Born again neural networks. *arXiv preprint arXiv:1805.04770* .

Ganin, Y.; and Lempitsky, V. S. 2015. Unsupervised Domain Adaptation by Backpropagation. *ArXiv* abs/1409.7495.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. J. 2006. A Kernel Method for the Two-Sample-Problem. In *NeurIPS*.

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: 1735–1780.

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* abs/1909.10351.

Johnson, J.; Hariharan, B.; van der Maaten, L.; Hoffman, J.; Fei-Fei, L.; Zitnick, C. L.; and Girshick, R. B. 2017. Inferring and Executing Programs for Visual Reasoning. *ICCV* 3008–3017.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155* .

Li, Y.; Swersky, K.; and Zemel, R. S. 2015. Generative Moment Matching Networks. In *ICML*.

McAuley, J. J.; Leskovec, J.; and Jurafsky, D. 2012. Learning Attitudes and Attributes from Multi-aspect Reviews. *ICDM* 1020–1025.

Paranjape, B.; Joshi, M.; Thickstun, J.; Hajishirzi, H.; and Zettlemoyer, L. 2020. An Information Bottleneck Approach for Controlling Conciseness in Rationale Extraction. *arXiv preprint arXiv:2005.00652* .

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv* abs/1802.05365.

Radford, A. 2018. Improving Language Understanding by Generative Pre-Training.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv* abs/1910.01108.

Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825* .

Sun, B.; and Saenko, K. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 443–450. Springer.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3319–3328. JMLR. org.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. *ArXiv* abs/1706.03762.

Waibel, A. H.; Hanazawa, T.; Hinton, G. E.; Shikano, K.; and Lang, K. J. 1989. Phoneme recognition using time-delay neural networks. *ICASSP* 37: 328–339.

Wang, L.; and Yoon, K.-J. 2020. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *arXiv preprint arXiv:2004.05937* .

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J. G.; Salakhutdinov, R.; and Le, Q. V. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NeurIPS*.

Yim, J.; Joo, D.; Bae, J.; and Kim, J. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, 4133–4141.

Yoon, J.; Jordon, J.; and van der Schaar, M. 2019. INVASE: Instance-wise Variable Selection using Neural Networks. In *ICLR*. URL https://openreview.net/forum?id=BJg_roAcK7.

Yu, M.; Chang, S.; Zhang, Y.; and Jaakkola, T. S. 2019. Rethinking cooperative rationalization: Introspective extraction and complement control. *arXiv preprint arXiv:1910.13294* .

Zellinger, W.; Grubinger, T.; Lughofer, E.; Natschläger, T.; and Saminger-Platz, S. 2017. Central Moment Discrepancy (CMD) for Domain-Invariant Representation Learning. *ArXiv* abs/1702.08811.