Zero-Shot Rationalization by Multi-Task Transfer Learning from Question Answering

Po-Nien Kung Tse-Hsuan Yang Yi-Cheng Chen Sheng-Siang Yin Yun-Nung Chen
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
{b06902012, b06902032, b06902011, b06902103}@ntu.edu.tw y.v.chen@ieee.org

Abstract

Extracting rationales can help human understand which information the model utilizes and how it makes the prediction towards better interpretability. However, annotating rationales requires much effort and only few datasets contain such labeled rationales, making supervised learning for rationalization difficult. In this paper, we propose a novel approach that leverages the benefits of both *multi-task learning* and *transfer learning* for generating rationales through question answering in a zero-shot fashion. For two benchmark rationalization datasets, the proposed method achieves comparable or even better performance of rationalization without any supervised signal, demonstrating the great potential of zero-shot rationalization for better interpretability.[1]

1 Introduction

Resolving NLP tasks by deep neural networks has been proven to be effective, and it is also important to investigate how the models make such a decision. For example, only providing the prediction to medical tasks may not be enough, and providing the associated reasons is more crucial for the practical applications. Therefore, there has been increasing attempts that focus on *interpretability* or *explainability* of the machine-learned models. There are different ways of explaining how machines make the decision (Ribeiro et al., 2016; Alvarez-Melis and Jaakkola, 2017; Lee et al., 2019; Liu et al., 2019a), and one of these methods is to extract rationales (Lei et al., 2016; DeYoung et al., 2019). However, most prior work focused on extracting rationales in a supervised manner (DeYoung et al., 2019), but not all datasets contain such annotated rationales for model learning, making the rationalization task difficult and impractical.

Rationalization is defined as a task that focuses on extracting the rationales from the input texts for better justification and interpretation. Lei et al. (2016) is the first work that attempted to extract rationales in order to justify the model's answers, where a rationale generator extracts the context and a predictor generates the answer based on the extracted rationales. This method shows great precision in extracting rationales. Recently, Yu et al. (2019) proposed an introspective model, an extension of the prior work that further improved the comprehensiveness of the extracted rationales. Moreover, DeYoung et al. (2019) proposed to learn rationale extraction in a supervised manner and prepared the benchmark experiments in diverse rationalization tasks. From the experimental results, it can be found that supervised learning for rationalization may not be always better than the unsupervised method due to the complex reasoning process.

Considering that in the practical application, the target domain may not contain the annotated rationales for supervised training, transferring the knowledge about rationalization to the target domain may be applicable. Rajani et al. (2019) proposed to utilize the pre-trained language model for explaining the common sense towards zero-shot knowledge transfer. However, it requires that the target domain should be covered by the pre-trained language model so that the common sense questions can be well-answered. Such requirements limit the potential of being applied to a lot of real-world applications, because the target domain we aim at extracting rationales for may not be general (e.g. medical texts and financial texts may not be covered by the pre-trained model). Instead of directly transferring the knowledge to the target domain, this paper proposes to borrow the benefit of multi-task learning, which allows a single model to be capable of handling multiple tasks/domains

[1]The source code and the processed data is available at: https://github.com/MiuLab/ZeroShotRationale.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2187–2197

Corpus	Set	#Data	Avg Words	%Rationale
SQuAD 2.0	Train/Dev	130,000/1,000	116	–
BeerReview	Train/Dev	30,000/3,000	152	–
	Test	994	127	18%
MovieReview	Train/Dev	45,000/5,000	317	–
	Test	199	795	30%

Table 1: The detailed statistics for three datasets. **Avg Words** denotes the average number of input words per instance. **%Rationale** denotes the ratio of the word number of words in the rationale to that in the input. Note that SQuAD 2.0 and the Train/Dev set of Beer/Movie Review do not provide rationales.

by feeding the corresponding data. Specifically, we utilize question answering for learning the capability of rationalization instead of a rationalization-specific language model. We can use the data (with the labels different from rationales) from one domain to feed into our multi-task model such that the QA-part of our model is capable of performing rationalization on the target data from other domains.

To enable multi-task transfer learning towards zero-shot rationalization, focusing on gaining the insight into data and simultaneously maintaining the capability of generalization is not trivial. Multi-task learning can address the issue about lack of training data in certain domains and alleviate overfitting through regularization effect (Ruder, 2017). Furthermore, Liu et al. (2019b) showed that training on different tasks by turns in every batch can significantly boost the regularization effect towards better generalization. Following the prior success, this paper focuses on extracting rationales via the capability of QA and handling other tasks with the target data at the same time; with multi-task learning for enhancing the capability of generalization, the model can handle diverse questions (including rationalization) from diverse domains.

This paper has three-fold contributions:

- This paper is the first attempt that leverages multi-task learning for zero-shot rationalization.
- This paper transfers the capability of question answering to extract the rationales in a zero-shot manner, and provides the potential of answering diverse questions even without any annotated information.
- The experiments demonstrate that the proposed approach achieves comparable or better performance than the prior work for two benchmark rationalization datasets without any annotated rationales.

2 Datasets

This paper focuses on zero-shot rationalization by transferring knowledge from question answering. Thus, three datasets are used in the experiments, where a QA dataset, **SQuAD 2.0**, is utilized for the transfer purpose and two benchmark rationalization datasets, **BeerReview** and **MovieReview**, are used for evaluating the performance of zero-shot rationalization for the proposed method. The datasets are briefly introduced below, and their statistics is detailed in Table 1.

SQuAD 2.0 Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) is a benchmark dataset for reading comprehension, consisting of questions posed by crowdworkers on over 500 Wikipedia articles. The answer to each question is a text span from the corresponding paragraph. There are 100,000 answerable questions and over 50,000 unanswerable questions similar to answerable ones written by crowdworkers adversarially. This data is for enhancing the ability of text understanding in our model so that we can transfer the knowledge to zero-shot rationalization.

BeerReview This is a beer review dataset processed by Lei et al. (2016)[2], which contains 1.5 million reviews written by the website users. The reviews have the associated multi-aspect ratings from 0 to 1: appearance, aroma, palate, taste, and overall rating in order. We randomly sample 30,000 reviews as our training set shown in Table 1. In addition, McAuley et al. (2012) provided sentence-level *annotated rationales* on 994 reviews, where each annotated sentence has its aspect label (one or multiple aspects), indicating what aspect this

[2] http://people.csail.mit.edu/taolei/beer/

sentence covers. These annotations can be seen as the rationale of the aspect-specific rating, which can be used for evaluating the extracted rationales.

MovieReview This dataset contains the reviews obtained from IMDB, where each review was labeled as positive or negative without any rationales, because it is originally proposed for sentiment analysis (Maas et al., 2011). Another similar dataset consists of 2,000 movie reviews from IMDB with their rationales that explain why the review is positive/negative (Zaidan et al., 2007). Note that each review may contain multiple rationales. Hence, we similarly utilize the annotated rationales as the testing data for validating the performance of zero-shot rationalization.

3 Proposed Approach

In order to perform zero-shot rationalization, we leverage the question-answering ability for finding the rationales in a given document that may come from an unseen domain. Here we propose an encoder-predictor model with multi-task learning illustrated in Figure 1, where the weights of the encoder are shared across different tasks (QA, beer rating classification, and movie rating classification). The multi-task learning model is to learn good representations of the inputs from different domains. To prevent the encoder from identifying the task type according to the input format, we add an additional question after each review as the new input, so that all inputs of three tasks are in the context-question format. In addition, there is a task-specific predictor added after the encoder for each distinct task, so they would not intervene with one another while training. During training, we fine-tune the pre-trained model for three tasks at the same time as illustrated in Figure 2. During testing, given a context from any domain with a corresponding question, the question-answering module (the right branch) is capable of finding the associated rationale we expect without training on the rationales from the target domain (highlighted in red in Figure 2), achieving zero-shot rationalization.

3.1 Model Architecture

In order to leverage the capability of multi-task learning, we construct a shared encoder and multiple task-specific predictors detailed below.

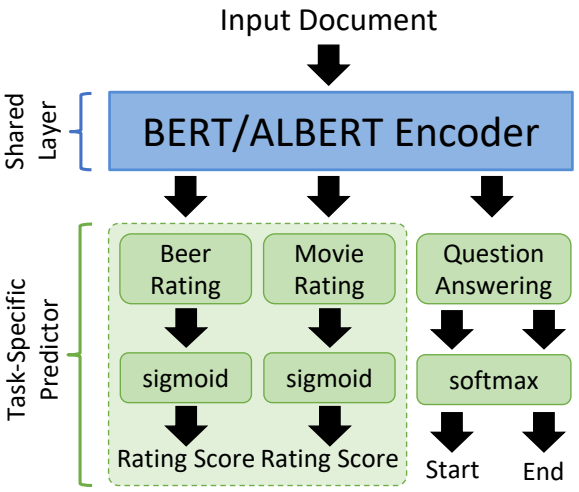

Figure 1: The proposed model architecture.

3.1.1 Shared Encoder

To utilize the universal understanding of the context, the pre-trained models are adopted. Here we chose **ALBERT** (Lan et al., 2019) as the encoder model considering its strong performance and simplicity. For each task, given the input (c, q), where c is the context with special tokens [CLS] at the start and [SEP] at the end, and q is the question with [SEP] at the end, the encoder $enc(c \oplus q)$ outputs a list of encoded vectors e:

$$e = enc(c \oplus q) = \{e_0, e_1, ..., e_n\}, \quad (1)$$

where $\oplus$ means concatenation, n is the length of input tokens, e_0 contains the condense meaning of the whole context, and e_i is the encoding of the i-th token in c .

3.1.2 Task-Specific Predictor

For each task, there is a corresponding predictor illustrated in each branch of Figure 1.

Question Answering For the QA task, we follow the implementation in Devlin et al. (2018) to construct the predictor $pred_{qa}(\cdot)$ with two dense layers, one for the answer start position and another for end.

$$v_s[i], v_e[i] = pred_{qa}(e_i), \quad (2)$$
$$a_s = \arg\max(\text{softmax}(v_s)), \quad (3)$$
$$a_e = \arg\max(\text{softmax}(v_e)), \quad (4)$$
$$y_{qa} = c[a_s : a_e]. \quad (5)$$

Hence, we can obtain the answer span y_{qa} based on the predicted answer start a_s and answer end a_e.

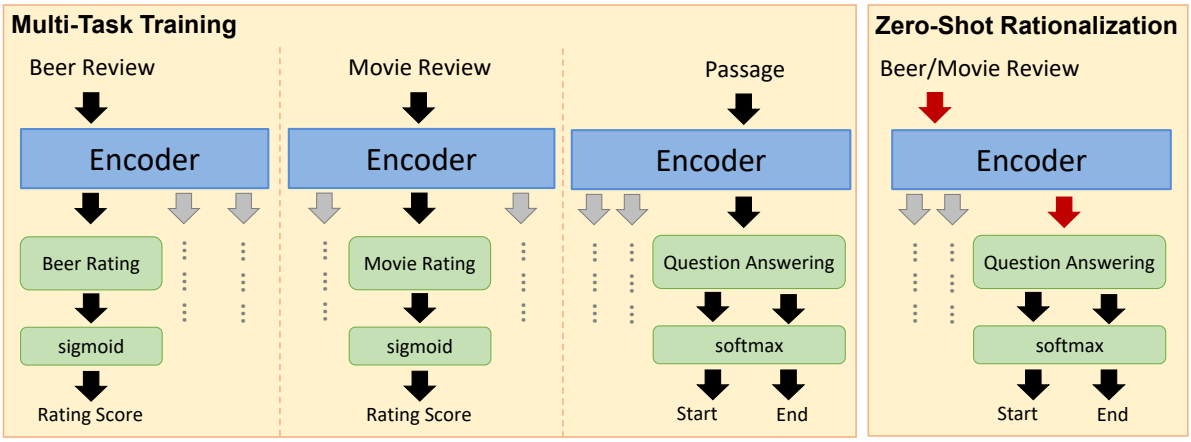

Figure 2: The illustration of the proposed multi-task training procedure and zero-shot inference for rationalization.

Beer Rating For the bear rating task, we construct the predictor $pred_{beer}(\cdot)$ using a dense layer, which inputs the encoding vector e_0 and outputs a value v_{sent}. e_0 is the embedding of the [CLS] token containing the condense meaning of the whole input.

$$v_{sent} = pred_{beer}(e_0), \quad (6)$$
$$y_{beer} = \text{sigmoid}(v_{sent}), \quad (7)$$

where y_{beer} is the output sentiment value between $[0, 1]$ for the given beer comment associated with a specific aspect.

Movie Rating Different from the eleven-level rating in the beer rating task, we formulate the sentiment analysis task into a rating prediction task, where "positive" and "negative" indicate 1 and 0 respectively, considering that the rationales are not well-associated with the fine-grained rating. The predictor structure is the same as one in the beer rating task, where a dense layer inputs e_0 and outputs one value v_{sent} to form the rating score y_{movie}. Here the output is considered as positive if $y_{movie} > 0.5$; otherwise negative.

3.2 Training Process

To enable multi-task learning, we control the input documents with the same format and apply alternative training for model training.

3.2.1 Input Formulation

In order to avoid the model from distinguishing the task based on the given text format, we reform the input data such that the input formats from three tasks are the same described below.

Input Context We construct the context into the format starting with [CLS] and ending with [SEP]:

$$c = \text{[CLS]} \oplus \text{Context} \oplus \text{[SEP]}. \quad (8)$$

Input Question Because two rating tasks only contain contexts and target ratings, the natural language questions are constructed based on the predefined templates. For example, each bear rating sample has the question "**What is this beer [appearance/aroma/palate/taste] score?**", and movie rating sample has the question "**How was this movie rated, positive or negative?**". Furthermore, to be consistent with the format as QA, the [SEP] token is appended with each question:

$$q = \text{Question} \oplus \text{[SEP]}. \quad (9)$$

3.2.2 Alternative Training

We train multiple tasks (QA and rating prediction) together by sharing the same encoder and using the corresponding predictor. In order to make the encoder generalize to all tasks, we train each task alternately for k turns in an epoch. That is, as illustrated in Figure 2, three parts representing three task training are equally considered during the multi-task training stage.

A corresponding objective is designed for each task. For QA, the ground truth target $\hat{y}_{qa} = (\hat{v}_s, \hat{v}_e)$ is constructed, where $\hat{v}_s$ and $\hat{v}_e$ are two binary vectors with only one element set to 1 and all others set to 0, and the only non-zero element in each vector indicates the answer start/end position. The cross entropy loss is applied to make the predicted start vector v_s close to the gold start position $\hat{v}_s$ ($\mathcal{L}_s$) and v_e close to $\hat{v}_e$ ($\mathcal{L}_e$). The overall loss is defined as $\mathcal{L}_{qa} = \mathcal{L}_s + \mathcal{L}_e$.

For two rating tasks, the target output $\hat{y}_{beer} \in \{0, 0.1, 0.2, ..., 0.9, 1.0\}$ is the beer rating score in a specific aspect, and the target output $\hat{y}_{movie} \in \{0, 1\}$ is the movie rating score. The MSE loss ($\mathcal{L}_{beer/movie}$) is utilized to make the predicted output $y_{beer/movie}$ approximate the target $\hat{y}_{beer/movie}$.

3.3 Zero-Shot Rationalization

In order to extract rationales in a zero-shot setting, we simply feed the context and question into the encoder and use the QA module to output the span of its rationale as illustrated in the right part of Figure 2. Specifically, both beer rating and movie rating tasks do not contain the labeled rationales during training, but our model is capable of extracting their rationales by transferring the knowledge from QA. Note that when extracting rationales from these two tasks, different post-processing methods are applied to control the format and length of the extracted rationales. The post-processing algorithm is detailed in Appendix A.

4 Experiments

In our rationalization task, we compare the performance with the prior work in terms of precision and recall on BeerReview and token F1 and IOU F1 on MovieReview.[3] Furthermore, we show the F1 scores of our model on BeerReview data for future benchmark comparison.

4.1 Settings

As mentioned in Section 2, we use the training set from SQuAD 2.0 (Rajpurkar et al., 2016), 30,000 BeerReview samples (Lei et al., 2016), and 45,000 MovieReview samples (Maas et al., 2011) for multi-task training. All input contexts and questions are truncated to the max length 384. During training, we use Adam (Kingma and Ba, 2014) as our optimizer with the learning rate of $5e-6$ and use ALBERT-Large (Lan et al., 2019) as the encoder structure in our model. We train each model for 3 epochs with the batch size 12 and tune the hyper-parameters[4] based on the dev set.

4.2 Evaluation Metrics

To evaluate the quality of the extracted rationales, two metrics, Intersection-Over-Union (IOU) F1 (Everingham et al., 2010; DeYoung et al., 2019) and token F1, are computed.

[3]The chosen metrics are consistent with the prior work for fair comparison.

[4]See Appendix B.3 for detail.

IOU F1 For a predicted rationale span p and a ground-truth rationale span a, we define the size of their union ($p \cup a$) as the number of all tokens (without computing mutual tokens repeatedly).

$$|p \cup a| = \max(|T \text{ in } p|, |T \text{ in } a|)$$
$$T = \{t \mid t \in \text{bag of tokens in } (p + a)\}.$$

Additionally, we define the size of intersection ($p \cap a$) as the number of matched tokens. Then, the IOU score between p and a is defined as

$$\text{IOU} = \frac{p \cap a}{p \cup a}. \tag{10}$$

For each prediction p, we find the maximum IOU score among those as for the same instance. We count p as a match if the maximum IOU score ≥ 0.5. Hence, we can compute IOU precision, recall, and F1 according to the number of matches, predictions, and ground-truth answers.

We have the word number of matched predictions, the word number of predictions and the word number of ground-truth answers. The definition of the IOU precision and recall is defined as follows:

$$\text{IOU Precision} = \frac{\text{\# of matches}}{\text{\# of predictions}}, \tag{11}$$

$$\text{IOU Recall} = \frac{\text{\# of matches}}{\text{\# of answers}}. \tag{12}$$

Based on the precision and recall, we can compute the IOU F1 score. This measure is more suitable for evaluating the results with multiple outputs in a single instance.

Token F1 The metrics is widely used for QA tasks, which assigns each rationale an F1 score. For both rating tasks, we choose the maximum F1 score for each prediction according to their answers of the same instance, because they may have multiple rationales in the same instance.

We compute the F1 scores as macro F1. That is, we first compute F1 score for each instance as the average F1 among all predictions in the instance. Then the overall F1 score is the average of all instance-level F1.

4.3 Beer Rating Rationalization

Baselines In the experiments, we compare our model with two baselines, Lei et al. (2016) and Yu et al. (2019). The previously proposed cooperative method was proven to have great performance on extracting rationales, where a generator and a predictor are built (Lei et al., 2016). The generator is

Method	10% Precision	20% Recall
Lei et al. (2016)	86.14	79.98
+ minimax (2019)	86.54	**85.16**
Intros (2019)	68.37	59.63
+ minimax (2019)	85.67	79.40
Only train on SQuAD	47.14	35.99
Proposed: S+B	92.13	75.65
Proposed: S+B+M	**93.41**	77.73

Table 2: Performance compared with the prior SOTA (S: SQuAD; B: BeerReview; M: MovieReview) (%).

Model	Prec Var.
Generator (Independent) (2016)	72.25
Generator (Recurrent) (2016)	80.44
Proposed: S+B	**0.56**
Proposed: S+B+M	8.36

Table 3: The precision variance among all aspects in beer reviews, showing the capability of generalization.

designed and trained to extract rationales, and the predictor is trained for rating prediction. Yu et al. (2019) further added a complement predictor and a target predictor to improve the comprehensiveness of extracted rationales, which significantly improved the recall score of extracted rationales. For fair comparison, we extract the same percentage of rationales from input contexts as these baselines shown in Table 2.

Results Table 2 shows the rationalization performance in terms of the precision when extracting 10% words as rationales and the recall when extracting 20% words as rationales in the "appearance" aspect[5]. All models are compared under the same condition, extracted 10% and 20% words as rationales compared to the gold-standard rationales from the context. It can be found that our proposed model outperforms all prior work when extracting 10% words as rationales and obtains good performance when extracting 20% words. However, Lei et al. (2016) with the additional complement predictor (+minimax) proposed by Yu et al. (2019) achieves the best recall for 20% results. It is reasonable because the additional complement predictor is to ensure the comprehensiveness and then the recall can be further improved. The results in Table 2 demonstrate that our model successfully transfers the capability of rationalization acquired from QA to perform on the beer domain in a zero-shot manner. Furthermore, Table 3 shows the precision variance among three aspects (appearance, aroma, and palate) of the baseline methods and the proposed model, where the percentages of extracted rationales are the same in all models for fair comparison. The larger variance of baselines is due to rationalizing-specific training (Lei et al., 2016), which may cause instability when extracting rationales. In contrast, our model utilizes the generality from multi-task learning, which is expected to extract rationales in a more stable manner (lower variance), indicating that the proposed method generalizes to different aspects better than baselines.

Table 4 shows the detailed scores for multiple aspects in beer reviews to benchmark the performance for future comparison. However, unlike cooperative models that train a rationalizing-specific structure (the generator), our proposed model simply applies the technique of task-transfer learning and extracts rationales using the generality of the encoder through question answering. This means that our model not only extracts rationales from the trained domains but also answers other question types requiring comprehension. The further discussion is detailed in Section 5. When only training on SQuAD, the model cannot achieve good performance for all cases, which tells that the knowledge from SQuAD cannot be directly utilized in the target domain due to domain mismatch. By leveraging multi-task learning, the proposed model is capable of extracting reasonable rationales from the target domain even though the training data does not contain any labeled rationales to learn from, demonstrating the effectiveness of zero-shot transfer through multi-task learning. In addition, comparing between the proposed models (S+B and S+B+M), the one training with two rating tasks (S+B+M) obtains better performance than the one trained without movie rating (S+B), showing that our model can extract some knowledge or commonsense by using the movie data and successfully transfers the domain knowledge to help extract the rationales in beer reviews.

4.4 Movie Rating Rationalization

Baselines For movie reviews, we compare our model with the baselines provided by DeYoung et al. (2019). They implemented a Bert-To-Bert model and the model in Lehman et al. (2019), both of which directly learn from the labeled rationales

[5]The prior work only performs on a single aspect.

Model	Appearance				Aroma				Palate			
	P	R	F1	IOU F1	P	R	F1	IOU F1	P	R	F1	IOU F1
20% Selected as Rationales												
Only train on S	46.0	31.5	24.3	16.2	68.8	51.3	37.7	27.2	62.0	33.2	18.4	11.5
Proposed: S+B	80.5	75.7	72.4	70.4	72.1	81.6	69.2	65.7	60.7	79.7	60.8	50.3
Proposed: S+B+M	82.4	77.7	74.5	72.6	72.8	80.1	69.2	64.8	58.2	76.3	57.7	48.7
10% Selected as Rationales												
Only train on S	45.2	19.1	15.8	8.35	88.8	30.5	34.0	22.8	66.4	31.3	20.6	17.1
Proposed: S+B	92.2	49.2	57.3	39.4	87.8	55.4	58.2	45.4	83.2	68.6	66.3	59.8
Proposed: S+B+M	93.6	52.5	61.7	46.1	88.2	53.5	57.4	42.6	81.0	68.5	64.4	56.6

Table 4: Detailed performance of zero-shot rationalization for three aspects in beer reviews (%).

in a supervised fashion. Note that our proposed method utilizes multi-task transfer learning to perform rationalization in an unsupervised manner.

Results Table 5 shows the results of movie rationale extraction. We compare all models using Token F1 and IOU F1 for fair comparison (DeYoung et al., 2019). Based on the results, it is shown that our proposed models all outperform two supervised baselines with large margins. It is surprising that without using any annotated rationales, our method can achieve remarkable performance in a zero-shot setting. In addition, only training the QA model with SQuAD can obtain the similar performance with two supervised baselines. It may be due to the small size of movie reviews annotated with rationales; hence, supervised learning for rationalization is relatively challenging. We find that the annotated rationales contain the information unnecessary for sentiment analysis (e.g. long movie plot descriptions without sentiment), and poor quality of the annotated rationales in movie reviews also leads to overall low accuracy. Therefore, cleaning annotations or finding other datasets with better quality is our future work.

With additional movie rating or/and beer rating tasks for multi-task training, our model significantly improves the performance for all cases even the data may not be relevant to the target data (BeerReview). The best model is the one trained with all three datasets, indicating that the rationalization ability of our model has the potential of being further improved by transferring the knowledge from other irrelevant data/tasks. The potential gives the future flexibility of different tasks performed in a zero-shot setting, demonstrating the impact of the proposed method.

5 Discussion

To better understand the limits and potential of our proposed method, we further study about the QA ability and comprehensiveness of our model.

Method	IOU F1	Token F1
Lehman et al. (2019)	6.3	13.9
Bert-To-Bert (2019)	7.5	14.5
Only train on SQuAD	6.2	15.3
Proposed: S+M	7.8	16.6
Proposed: S+B	8.0	18.5
Proposed: S+B+M	**9.3***	**19.6***

Table 5: Rationalization performance in movie rating (S: SQuAD; B: BeerReview; M: MovieReview) (%).

5.1 Diverse Question Types

Considering that the proposed model is trained with both rating tasks, other question types in addition to "why" are also likely to be answered. We further study the capability of text comprehension in our model by comparing the results from the proposed model and the simple QA model only trained on SQuAD, so that the task-transfer ability from rating prediction to QA can be investigated. Their results are shown in Table 6.

In Table 6, the answers outputted from two models are compared, where one is the model only trained on SQuAD and another is the one additionally utilizing BeerReview and MovieReivew data in multi-task training. Two questions are asked to the QA models: 1) an *abstractive* question related to sentiment and 2) an *extractive* question, where the answer is more precise and can be directly extracted from the context. For the abstractive question, it can be seen that our model gives the prediction about aroma, which is rated below average by the writer. As for the simple QA model, it predicts a sentence related to appearance, which has an average score. The difference shows that our model can answer an abstractive question related to sentiment better than a simple QA model. For the extractive question, the answer is exactly

Context	To be perfectly honest this is an average dunkel, fairly solid in all categories but average for the style. I have enjoyed this beer in the past and I continue to enjoy drinking some of these from time to time. Pours very dark for a dunkel and produces a thick foamy head that ultimately turns to a thin film. Minimal lacing slowly creeps down the glass and I am not able to determine the carbonation since I can not see through my glass. The aroma is malty, sweet, and bitter. I gave it a below average rating only because the aroma is very hard to detect, I'm forced to shove my nose down into my glass in order to pick up any scent. The flavor is definitely based on the original lager, but it appears to be altered with some sweet malts. The combination is very good and I sure do enjoy the overall taste of this one. The mouthfeel is decent not too thin and the finish is clean and smooth. All in all a well rounded beer for this style.
Question 1	Which part of this beer is considered bad? Only train on S, Proposed: S+B+M
Question 2	What did the writer do to get the smell? Only train on S, Proposed: S+B+M

Table 6: The rationales extracted from different Model.

Input	**Appearance**	**Aroma**	**Palate**
Rating Accuracy			
Full-context	64.9	65.5	61.5
Comprehensiveness (Δ Accuracy)			
w/o R_{random}	–2.9	–3.1	–1.8
w/o R_{S}	–3.8	–4.9	+0.7
w/o $R_{\text{S+B+M}}$	–6.7	–7.4	–6.9

Table 7: Comprehensiveness analysis of the extracted rationales from BeerReview. The performance is based on 5-level via scaling. *Comprehensiveness* shows the performance gap of the model fed with full input context and without model-extracted rationales. (%).

a sequence of words in the context. This question type is similar to questions in SQuAD, so the simple QA model is capable of predicting a short and precise answer. However, our model provides a longer span in the context. The probable reason is that multi-task learning focuses on improving the generality among all tasks/data; thus, the model tends to give longer answers that can generalize to many scenarios and may contain better comprehensiveness.

5.2 Comprehensiveness

We further investigate the comprehensiveness of our model, and we use beer data for analysis. The comprehensiveness is to evaluate whether the extracted rationales comprehensively include the salient rationales the model needs to produce accurate prediction (DeYoung et al., 2019). To evaluate the comprehensiveness, we start with training a BERT-based sentiment analysis model on the beer dataset. After the model is trained, we compare its accuracy of 5-level rating[6] prediction when using the full context as inputs and ones removing the extracted rationales.

Table 7 shows that when randomly removing the extracted rationales (w/o R_{random})[7] or by simple QA model trained on SQuAD only (w/o R_{S}), the performance is slightly lower than the one with the full context. When removing the rationales extracted by our proposed model (w/o $R_{\text{S+B+M}}$), the performance significantly drops with a great margin. The results prove that our model not only extracts rationales with higher precision but also preserves good comprehensiveness.

6 Conclusion

In this paper, we propose a novel framework that leverages multi-task learning for zero-shot task transfer, where the question answering model is utilized to perform rationalization for diverse domains. By training on multiple tasks alternately, we improve the universal understanding of the context and is able to use the QA structure to extract rationales from any tasks/data. The experiments of benchmark rating prediction datasets for two domains are conducted, and the results show that our proposed model achieves comparable or better performance of rationalization compared with the prior work and meanwhile preserves better capability of generalization and flexibility towards better interpretability.

[6]The original task of the beer dataset is an 11-level sentiment analysis task. To better show the results of comprehensiveness, we modify the rating task from 11-level to 5-level to create a significant difference between ratings.

[7]R_{random} is a random span with the same length as the extracted rationales by the proposed model for fair comparison.

Acknowledgments

We thank reviewers for their insightful comments. This work was financially supported from the Young Scholar Fellowship Program by Ministry of Science and Technology (MOST) in Taiwan, under Grant 109-2636-E-002-026.

References

David Alvarez-Melis and Tommi Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 412–421.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Guang-He Lee, Wengong Jin, David Alvarez-Melis, and Tommi Jaakkola. 2019. Functional transparency for structured data: a game-theoretic approach. In *International Conference on Machine Learning*, pages 3723–3733.

Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. *arXiv preprint arXiv:1904.01606*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

Hui Liu, Qingyu Yin, and William Yang Wang. 2019a. Towards explainable nlp: A generative explanation framework for text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5570–5581.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Julian McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. In *2012 IEEE 12th International Conference on Data Mining*, pages 1020–1025. IEEE.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi S Jaakkola. 2019. Rethinking cooperative rationalization: Introspective extraction and complement control. *arXiv preprint arXiv:1910.13294*.

Omar F. Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *NAACL HLT 2007; Proceedings of the Main Conference*, pages 260–267.

A Post-Processing

For beer rating and movie rating tasks, two post-processing algorithms are applied based on their data nature. For example, rationalization data may contain multiple rationales, where the output format is different from the QA task (one answer for each question in SQuAD 2.0). Both post-processing algorithms are expansions from the method used in Devlin et al. (2018) for the SQuAD 2.0 dataset.

SQuAD 2.0 The post-processing method for SQuAD 2.0 dataset used in Lan et al. (2019) is simply using the a_s, a_e in (3) and (4) as the extracted answer span, where each is the index of the highest value in the vector v_s or v_e in (2). For unanswerable questions, the post-processing method set a answer to an empty string (indicating no answer) if the probability of "[CLS]" in both $\text{softmax}(v_s), \text{softmax}(v_e)$ exceeds a predefined threshold ϵ or $a_e < a_s$, which indicates an invalid answer span.

$$P_{cls} = \text{softmax}(v_s)[0] + \text{softmax}(v_e)[0]$$

$$y_{qa} = \begin{cases} \text{No Answer}, & \text{if } P_{cls} < \epsilon \text{ or } a_e < a_s \\ c[a_s : a_e], & \text{otherwise.} \end{cases}$$

where P_{cls} is the probability of "[CLS]" and y_{qa} is the final answer. However, in our implementation, we do not check if P_{cls} exceeds ϵ but simply check if one of a_s, a_e is zero, which means that the "[CLS]" token has the highest probability to be the start or end indices of the answer span. Thus the post-processing method becomes:

$$y_{qa} = \begin{cases} \text{No Answer}, & \text{if } a_s = 0 \text{ or } a_e = 0 \\ & \text{or } a_e < a_s \\ c[a_s : a_e], & \text{otherwise.} \end{cases}$$

Beer-Rating To extract rationales from beer data, a post-processing algorithm is presented to control the length of the outputted rationales, which iteratively expands the answer span until it matches the threshold we set. This algorithm is the expansion from the previous implementation described in the above section. The pseudo code for the implementation is detailed in Algorithm 1.

In our method, if $a_e < a_s$ occurs, we find a new a_s or a_e to make the answer span valid in order to reduce the amount of misjudgment unanswerable prediction, because this condition is more likely to happen when predicting rationales due to the nature of longer answers. When expanding answer spans, we iteratively find the new answer-start and answer-end index with the second highest probability, until the length ratio of the answer in context exceeds the threshold we set. By scaling the threshold ϵ, we calculate the average length of all predicted rationales and divide it with average context length. The result will be the highlighting ratio, the portion of rationale we extracted from input contexts.

Algorithm 1 Post-processing for extracting rationales from BeerReview data

```
// C is the input context, Q is the input question
Input C, Q
v_s, v_e := pred_qa(Encoder(C, Q))
v_s ← softmax(v_s)
v_e ← softmax(v_e)
a_s = arg max(v_s)
a_e = arg(v_e)
threshold_ε := |C| * ε
while a_e − a_s < threshold_ε do
    a_s^new = a_s + arg max(v_s[: a_s])
    a_e^new = a_e + arg max(v_e[a_e :])
    if |a_s − a_s^new| < |a_e − a_e^new| then
        a_s ← a_s^new
        if |a_e − a_s^new| > threshold_ε then
            break;
        end if
    else
        a_e ← a_e^new
        if |a_e^new − a_s| > threshold_ε then
            break;
        end if
    end if
end while
Answer := C[a_s : a_e + 1]
```

Movie-Rating In the movie-rating task, the main difficulty is that most input contexts have multiple human-labeled rationales, where our model can only output one answer span with an input context. To resolve the problem, we split the input context to a list of sentences in which each of them is fed into our model as a complete context. After all sentences in a context were predicted, we then combine the outputs and acquire the complete rationale.

B Reproducibility

To reproduce the model and the evaluation results, we provide the detailed settings of our experiment.

B.1 Training details

We use a **Tesla P40 GPU** and **Intel(R) Xeon(R) CPU E5-2667** to train our model. Each epoch takes about **15 hours** with a **batch size 12** for the SBM model. We evaluate each model with the valid loss of beer-rating, movie-rating and SQuAD dataset, the best model(lowest valid loss on all three tasks) are usually happening in the first or second epoch. When alternately train the three tasks, we divide each dataset into ten parts and train for ten turns, in the order of Beer, Movie, SQuAD. For loss function, we use **MSELoss** for beer-rating and movie-rating tasks and use **CrossEntropyLoss** for SQuAD. The learning rate is set to **5e-6**, using Adam as an optimizer without a warm-up step.

B.2 Model details

We use PyTorch and transformers package to implement our method. The model is constructed using **Albert(albert-large-v2)** as Encoder and three linear layers as the three task-oriented predictors. The total parameter number of our model is **17.6M**. The best SBM model was trained for **3 epochs**, with valid losses **0.94, 0.059, 0.043** for SQuAD, beer-rating, and movie-rating tasks respectively.

B.3 Hyperparameters

Our model was trained with hyperparameter search, where we tested the alternatively training-step with 10 steps and 100 steps and found out that 10 steps perform slightly better on validation. For the learning rate, we found that when using $1e-5$ without warm-up, the valid loss of SQuAD is 10% higher than using $5e-6$. To maintain a better balance when training three tasks together, we multiply the loss of beer-rating and movie-rating tasks by 10.

B.4 To Reproduce

We provide the code and data needed for reproducing our proposed SBM model. To reproduce, download the appendix software and data, and follow the instructions in **Reproduce.md** stored in the software directory.