ABSTRACT
A desirable property of learning systems is to be both effective and interpretable. Towards this goal, recent models have been proposed that first generate an extractive explanation from the input text and then generate a prediction on just the explanation called explain-then-predict models. These models primarily consider the task input as a supervision signal in learning an extractive explanation and do not effectively integrate rationales data as an additional inductive bias to improve task performance.

We propose a novel yet simple approach ExPred, which uses multi-task learning in the explanation generation phase effectively trading-off explanation and prediction losses. Next, we use another prediction network on just the extracted explanations for optimizing the task performance. We conduct an extensive evaluation of our approach on three diverse language datasets – sentiment classification, fact-checking, and question answering – and find that we substantially outperform existing approaches.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM ’21), March 8–12, 2021, Virtual Event, Israel. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441758
effectively utilize the rationales data as a supervision signal. Specifically, Bastings et al. [2], Lei et al. [19], Yoon et al. [36] train end-to-end models that are only supervised on task-specific training data. On the other hand, Lehman et al. [18] follows a pipelined approach that explicitly uses the rationales data in the explanation generation phase but is agnostic to task-specific signals, thus not being able to generalize well in the subsequent prediction phase. This paper’s main objective is to exploit supervision signals from both rationales data and task objective for generating task-aware explanations to improve task performance.

Unlike earlier approaches, our idea is simple – we learn to generate explanations supervised by both task-specific and rationale-based signals in our explanation generation phase. We realize this by using multi-task learning, where task prediction and explanation generation are both learned on a common encoder substrate (cf. Figure 2). After training the explanation model, in the prediction phase, a separately parameterized model for task prediction is learned just on the generated explanation. We refer to this scheme of predicting and explaining first (in the explanation generation phase) and then predicting again (in the prediction phase) as ExPred.

We conduct an extensive evaluation of ExPred on three different language tasks found in the Web, where human rationales are provided – sentiment classification, fact checking and question answering. We find that using a shared representation space for encoding the input for prediction and explanation generation results in more task-specific explanations. We also observe that ExPred can effectively balance the task and explanation performance by learning to generate task-specific explanations.

Our contributions. In sum, the key contributions of our work are

- We propose a novel explanation generation framework work using multi-task learning ExPred that is task-aware and can exploit rationales data for effective explanations.
- We show that our explanations show significant improvements in task performance (up to 7%) and explanation accuracy (up to 20%) over existing baselines.

For the sake of reproducibility, the code for the experiments described in this paper will be made available at https://github.com/JoshuaGhost/expred.

2 RELATED WORK

Classical models are known to exhibit a natural trade-off between task performance and being interpretable. As a result, in recently popular post-hoc interpretability approaches that do not negotiate task performance and instead rely on interpreting already trained models in a post-hoc manner [16, 22, 26, 34]. However, a fundamental limitation of such post-hoc approaches is that the explanations might be faithful to the predictions of the model but might not be faithful to the model’s actual decision-making process of the model [28] or to human reasoning [40]. Secondly, and more worrysome is the problem of evaluation of interpretability techniques due to difficulty in gathering ground truth for evaluating an explanation due to human bias [17].

Explain-then-predict models. Lei et al. [19] proposed a sequential approach of rationale generation followed by prediction using the generated prediction. Similar frameworks that mainly differ in how they perform end-to-end training due to the explanation sampling step have been proposed subsequently. Common proposals for training include using REINFORCE [19], actor-critic methods [36], or re-parameterization tricks [2], Lehman et al. [18] uses a similar philosophy of decoupling rationale generator and predictor, albeit a slightly different architecture and supervised using human rationales. Instead, we explicitly use human rationales to provide the supervision signal and decouple the prediction network from the explanation phase.

Rationale-based prediction. Related to our work is the work on rationale classification and has roots in the seminal work of Zaidan et al. [37], Zaidan and Eisner [38] that aims to improve model generalization by utilizing human rationales as inductive bias. The closest to our work of building explain-then-predict models using rationale data is DeYoung et al. [7], who instead use rationale predictions as regularizers to the task loss. We use these approaches as competitors in our experiments.

Unlike us, all these approaches are agnostic to task supervision when learning to generate explanations. An exception is Zhong et al. [41] that showed supervising regularizes the attention layer with human annotations while learning from task supervision. However, it is not an explain-then-predict model. Specifically, it is hard to unambiguously attribute the rationale of the prediction since the prediction phase still has access to the input.

Interpretability for Language Tasks. With the tremendous growth of the Web [10, 11], many language tasks on the Web are being treated learning tasks. For language tasks, there has been work on post-hoc analysis of already learned neural models by analyzing state activation [1, 9, 20] or attention weights [5, 12, 23, 35]. The attention weights learned as weights assigned to token representation are intended to describe rationales. However, recently the faithfulness of interpreting model prediction with soft attention weights has been called into question [13, 33]. Specifically, the contextual entanglement of inputs is non-trivial. The prediction model can still perform well even if the attention weights don’t correlate with the (sub-)token weight as desired by humans. Our approach for rationale based explanations differs in the type of architectures, objectives, and general nature of its utility.
3 APPROACH
We aim to come up with a model that can generate explanations as well as high-quality predictions, given access to human rationales accompanying task-specific training instances. Human rationales are sets of sequences of the input text that have been annotated by humans as potential reasons for the prediction.

We formalize here the task of extractive rationale generation in the context of neural models where we are provided with a sequence of words as input, namely $x = (x^1, \cdots, x^{|S|})$, where $|S|$ is the length of the sequence and each $x^i \in \mathbb{R}^d$ denotes the vector representation of the $i$-th word and task labels $y$. Additionally, we also assume that each word $x^i$ has an associated Boolean label $t^i \in \{0, 1\}$, where $t^i = 1$ if word $i$ is a part of the rationale else $t^i = 0$. The rationales of the sequence is then $t \in \{0, 1\}^{|S|}$. Typically, rationales are sequences of words and hence a potential rationale is a sub-sequence of the input sequence. Note that multiple non-overlapping sub-sequences might exist for a given input text.

3.1 Approach Overview
Our goal is to construct a explain-then-predict model that is composed of an explanation generation network $g^\phi$ parameterized by $\phi$ and a prediction network $f^\theta$ parameterized by $\theta$. The explanation generation network $g^\phi$ first maps the input $x$ into an explanation mask $t$. Thereafter, the prediction network $f^\theta$ maps the masked input $x \circ t$ to the task output $y$.

Our key insight is that generating effective task-specific explanations, we would ideally want to be influenced by task-specific supervision along with rationale-specific supervision. Towards this, we propose to use the Multi-task Learning (MTL) framework [3] for the explanation generation phase. In MTL, the original prediction task is trained along with multiple related auxiliary tasks using shared or tied parameters [21] as a form of inductive transfer that might exist for a given input text.

3.2 Explanation Generation
In our explanation generation phase, we detail our architectural design choices for encoders and decoders and our loss function.

3.2.1 Shared Encoder. Since contextualized models like BERT [6] are now de-facto models like for representing text input, we use the BERT model as our shared encoder $\text{enc}(\cdot)$ between $g^\phi$ and auxiliary task predictor $f^\psi$. In principle, BERT can be replaced by any text encoding model as an encoder – LSTMs, other transformer-based encoders, etc. We follow the standard practices in handling text input in BERT. Specifically, a single sentence or sentence pair is fed to BERT based on the type of tasks. Sentence tokens, segments, and positional information are taken as inputs. Technically, for a single sentence task, this is realized by forming an input to BERT of the form $([CLS], <\text{ sentence}>)$ and padding each sequence in a mini-batch to the maximum length (typically 512 tokens) in the batch. Similarly, a sentence-pair task is realized by $([CLS], <\text{ sentence1}>, <\text{SEP}>, <\text{ sentence2}>)$ and the entire sequence is of maximum length 512 [6]. The final hidden state corresponding to the $[CLS]$ token captures the high-level representation of the entire text and other vectors represent the corresponding embeddings of the input tokens. Hence, we obtain a $512 \times 768$ dimensional representation of the input sequence where 512 is the maximum number of input tokens.

The working principle of recent auto-regressive language models is significantly better than word-based representations (word2vec) and long dependency modeling networks (RNN and LSTM)[6]. Word2vec models assume independence between words present in a sentence that does not hold. Contextual auto-regressive neural models such as BERT overcome that limitation. The model also works as a knowledge-base due to its pre-training over a large amount of unlabelled corpus [25]. On the other hand, LSTM based models were proposed to capture long term dependencies among words and overcome the problem of vanishing gradient. However, this scheme does not work for large paragraphs. BERT completely relies on self-attention instead of multiple gates. This increases the complexity quadratically but helps to capture the interaction between each pair of words.

3.2.2 Decoders. We reiterate that we use the original prediction task as the auxiliary task. We employ a simple MLP to map the encoded input $\text{enc}(x)$ to the task prediction $y$. The choice of explanation decoder, however, induces interesting design choices. One could in principle pose the generation task as a span detection task or token prediction task. In this work, we pose the explanation generation as an independent binary classification task over each of the input words. We apply a gated recurrent unit (GRU) over the sequence of output token representations of BERT to consider sequential dependencies among tokens. Then, token representations from the GRU are pooled to form word representations followed by a word-wise MLP. Figure 3 shows the diagram of our proposed approach for the single sentence task (e.g., sentiment detection). The same task and explanation generation approach is followed for sentence pair tasks (question-answering) where both the sentences are fed to BERT.

3.2.3 Loss function. The explanation loss is composed of individual losses incurred on each input word and can be written as

$$L_{exp} = \frac{1}{|S|} \sum_{i=1}^{|S|} |S_{ti}| \cdot \text{BCE}(p^t, t^i),$$ (1)
weights are inverse of the prior probabilities of each class within
where
This is necessary to maintain the overall structure of the input
text. Note that since we have a pipelined approach, errors in the
input to the prediction phase is the extractive explanation as a
text. Because of the input length restriction of BERT, here

\[
L_{loss} = L_{task} + \lambda L_{exp},
\]

where \( L_{loss} \) is the overall loss and \( L_{task} \) and \( L_{exp} \) are loss functions for the task and explanation respectively. \( \lambda \) regulates the importance of loss function between task and explanation.

A key challenge in explanation generation is the presence of sparse labels, i.e., the majority of the input words/tokens are not explanations. This leads to training issues due to the label imbalance that the loss function has to account for. To account for label sparsity, following Chawla et al. [4] we up-weight the log-likelihood of rationale, while calculating the binary cross entropy (BCE). The weights are inverse of the prior probabilities of each class within each input passage, i.e., the inverse proportion of non-rationale tokens in the passage.

### 3.3 Prediction Model

The input to the prediction phase is the extractive explanation as a masked input \( x \otimes g^\phi(x) \). Specifically, we replace each token that is not in the explanation with a wildcard token (period ‘.’ here). This is necessary to maintain the overall structure of the input text. Note that since we have a pipelined approach, errors in the explanation generation phase might lead to error magnification in the prediction phase. Towards this, rather than considering all input instances for training, we limit ourselves to input instances where the auxiliary task prediction is the same as the actual task label, i.e., \( f^\psi(x_i) = y_i \) for a training instance \( (x_i, y_i) \). We also choose BERT as the network \( f^\psi / f^\tau \) that aims to predict the true task label. The second-stage model is also validated on such masked inputs. But we don’t rule out any instance according to the auxiliary model prediction during the validation to reflect what happens during test time.

Mathematically, for an instance \( (x, t, y) \), the training function of ExpPred works as per equation. 3.

\[
\begin{align*}
    f^\psi(x) &\mapsto \{0, 1\} \\
    g^\phi(x) &\mapsto \{0, 1\}^{|S|} \\
    f^\psi(x \otimes g^\phi(x)) &\mapsto \{0, 1\}, \text{ if } f^\psi(x) = y
\end{align*}
\]

The inference is also similar but the output of the auxiliary task predictor is not taken under consideration (eqn. 4).

\[
\begin{align*}
    f^\psi(x) &\mapsto \{0, 1\} \\
    g^\phi(x) &\mapsto \{0, 1\}^{|S|} \\
    f^\psi(x \otimes g^\phi(x)) &\mapsto \{0, 1\}
\end{align*}
\]

### 4 EXPERIMENTAL EVALUATION

We first describe the experimental setup, baselines, and dataset details. In the next section, we elaborate on the experimental results in detail followed by further analysis.

#### 4.1 Datasets

We consider three diverse language tasks for our evaluation from the benchmark in DeYoung et al. [7]. All datasets are split in the same way as provided in the benchmark. Since we use BERT for representing inputs that have a natural length limitation, we refrain from experimenting with other datasets in the benchmark that contain longer sentences and might require non-trivial input segmentation. Extending our approach to documents with longer sentences and other datasets in the benchmark is left for future work.

**Movie Reviews** Zaidan et al. [37], Zaidan and Eisner [38]. One of the original datasets providing extractive rationales, the movies dataset has positive or negative sentiment labels on movie reviews. As the included rationale annotations are not necessarily comprehensive (i.e., annotators were not asked to mark all text supporting a label), DeYoung et al. collected a comprehensive evaluation set on the final fold of the original dataset [24].

**FEVER** Thorne et al. [32] (short for Fact Extraction and VERification) is a fact-checking dataset. The task is to verify claims from textual sources. In particular, each claim is to be classified as supported, refuted or not enough information with reference to a collection of potentially relevant source texts. We follow the setup of DeYoung et al. [7] who restricted this dataset to supported or refuted.

**MultiRC** Khashabi et al. [14]. This is a reading comprehension dataset composed of questions with multiple correct answers that by construction depend on information from multiple sentences. In MultiRC, each Rationale is associated with a question, while answers are independent of one another. We convert each rationale/question/answer triplet into an instance within our dataset. Each answer candidate then has a label of True or False.
4.2 Baselines, Competitors, Variants

We consider the following competitors that also use a pipelined approach to showcase the effectiveness of our approach:

- **Lei et al. [19]**: An end-to-end explain-then-predict approach where rationale generator and decoder are not supervised on rationales;
- **DeYoung et al. [7]**: An improvement of the approach of Lei et al. [19] where the final loss function has a regularizer based on rationale data. Note that this approach is denoted as Lei et al. (2016) and the previous one is denoted as Lei et al. (2016) (u) in [7];
- **Lehman et al. [18]**: It is a pipeline approach, where the explanation generation model is trained only on rationales, and the predictor model is trained on ground truth human rationales (instead of on machine predicted rationales as we do) as input to predict the task labels;
- **Bert-To-Bert**: It is implemented in [7], where the generator and the predictor are replaced by Bert followed by corresponding MLP heads. It is similar to our Expred (w/o Task Sup.) but we insert an additional GRU layer into the generator, i.e. after the Bert encoder of the explainer.

4.3 Metrics

Mostly denoted as Perf. in [7], the **Macro F1** produced by the classification_score from sklearn\(^1\) is used to evaluate task performance. Macro Token-wise F1, presented as **Token F1** in Table 1, is used to measure the proximity of the explanation with human rationales. The precision of an explanation is the fraction of commonly extracted rationale tokens (ER) and ground truth (GT) tokens in comparison to ER. While the recall of an explanation is the fraction of common ER tokens with GT in comparison to GT. The Token F1 is the harmonic average of precision and recall of machine rationales.

4.4 Training setup and Hyper-parameters

All experiments are conducted on an Nvidia 32GB V100 using the PyTorch and Tensorflow framework. We consider **BertBase** as the shared encoder model with MAX_SEQ_LEN = 512 and the warm-up proportion 0.1. Both the explanation generation and task prediction models are trained using Adam optimizer [15] with a batch size of 16, and learning_rate = 1e−5. Models are trained for 10 epochs with early-stopping criteria on the validation set and patience = 3. The MLP for the task classification consists of a dropout layer with a 10% chance of masking, followed by a 256 dimensional hidden dense layer, again followed by a Sigmoid output layer. The explanation decoder consists of a 128-dimensional GRU with a uniform random kernel analyzer. Note that the final outputs of the explanation generator correspond to the sub-token representations of Bert. Adjacent sub-tokens are merged to their corresponding original words through max-pooling. The best \(\lambda\) is chosen over a validation set that provides the best trade-off between task performance and token-F1. The best \(\lambda\) values for Movie Reviews, MultiRC, FEVER are 5.0, 20.0, 2.0 respectively. After training the explanation generation network in Expred, we remove instances that the auxiliary output predicts wrongly, and use the rest to train the prediction model. This is to avoid distraction from the wrong predictions from the explanation prediction phase. Note that this is only done during

\(^1\)https://scikit-learn.org/stable/

| Approaches                | Movie Reviews | FEVER | MultiRC |
|---------------------------|---------------|-------|---------|
|                           | Macro F1      | Token F1 | Macro F1 | Token F1 | Macro F1 | Token F1 |
| DeYoung et al. [7]        | 0.914         | 0.283  | 0.719   | 0.234   | 0.655    | 0.456   |
| Lei et al. [19]           | 0.920         | 0.322  | 0.718   | 1       | 0.648    | 1       |
| Lehman et al. [18]        | 0.750         | 0.139  | 0.691   | 0.523   | 0.614    | 0.140   |
| Bert-To-Bert              | 0.860         | 0.145  | 0.877   | 0.812   | 0.633    | 0.412   |
| Expred-Stage-1            | 0.884         | 0.348  | 0.907   | 0.837   | 0.718    | 0.640   |
| Expred (w/o Task Sup.)    | 0.814         | 0.142  | 0.795   | 0.801   | 0.725    | 0.609   |
| Expred                    | 0.915         | 0.348  | 0.894   | 0.837   | 0.698    | 0.640   |
| Human Explanation         | 0.899         | 1.0    | 0.921   | 1.0     | 0.759    | 1.0     |
| Full Input                | 0.894         | -      | 0.916   | -       | 0.708    | -       |

Table 1: Task and explanation performance of hard models, which is defined in section 5.2. Best performances , excluding the Token F1 of human annotation, since they are always 1.0, are bold and the second bests are underlined. Results for the competitors are kept the same as in the ERASER benchmark [7] whenever it is possible. Also according to [7], 1 indicates rationale training degenerated due to the REINFORCE style training.
training, while the predictions on the validation and test sets are regardless of the task prediction from the explanation phase.

5 RESULTS

We present the results of the effectiveness of our multi-task learning rationale generation framework in Table 1. Our first observation is that HUMAN EXPLANATION is quite effective in most datasets and MultiRC is significantly better than FULL INPUT in task performance. This is perhaps unsurprising because HUMAN EXPLANATION is trained on extractive rationales that contain task-specific discriminative tokens. This also suggests that FULL INPUT is sometimes distracted by words or tokens unrelated to the task and dropping terms altogether can result in reasonable task performance gains.

Among the variants of ExPred, ExPred (w/o Task Sup.) model is solely optimized on the explanation loss but has a moderate explanation quality. The explanation performance of ExPred and its variants are the best among all datasets and competitors. However, it does not perform better than HUMAN EXPLANATION in terms of task performance. This justifies our claim that purely optimizing for explanation accuracy without considering the task context leads to sub-optimal task performance. Note that ExPred-Stage-1 and ExPred generate the same explanation and have identical explanation quality since they both share the same explanation generation phase.

ExPred (w/o Task Sup.) is outperformed in explanation accuracy (in all datasets) and task accuracy (in Movie Reviews and FEVER) by ExPred-Stage-1 that jointly optimizes for the task and explanation using shared encoding parameters. For MultiRC and FEVER, both these variants are already much better than the competitors in task performance but seem less congruent with human rationales. A crucial difference between our variants with Lehman et al. [18] and Bert-To-Bert is that the prediction network for those two models is trained over human annotations. However, during the test phase, the output of the machine-generated explanation is considered.

Finally, we present the main result of our paper, i.e., ExPred and its variants convincingly outperform all other competitors in explanation performance by ~ 8% on Movie Reviews, ~ 5% on FEVER and more strikingly ~ 46% on MultiRC. Notably, the task performance is at least preserved on Movie Reviews or even improved on FEVER and MultiRC, compared with other competitors that use joint or rationale data-agnostic training. Comparing with Human Explanation further verifies our assumption that the models can learn more effectively from rationales data, where the right reasons of making predictions are highlighted in advance. We attribute this due to two reasons found in our earlier observations – as in the case of Human Explanation vs Full Input, ExPred being trained on sparser (less noisy) input can predict better. Secondly, the explanations are now more contextualized since they are learned along with the task. Furthermore, we can see that the task performance can even be sometimes slightly improved by adding a second classifier in the ExPred compared with the task prediction in the ExPred-Stage-1 (e.g. on Movie Reviews).

Table 1: Statistics of the machine-generated and human-annotated rationales. Precision and Recall are computed with respect to corresponding human-annotated explanations.

| Model Type | Avg. Rationale Len. | Precision | Recall |
|------------|---------------------|-----------|--------|
| Movie Reviews |                       |           |        |
| DeYoung et al. [7] | 8.533 | 0.662 | 0.033 |
| Lei et al. [19] | 430.563 | 0.315 | 0.542 |
| Lehman et al. [18] | 30.530 | 0.505 | 0.102 |
| Bert-To-Bert | 17.500 | 0.614 | 0.072 |
| ExPred (w/o Task Sup.) | 70.864 | 0.676 | 0.112 |
| ExPred | 86.246 | 0.607 | 0.284 |
| HUMAN EXPLANATION | 240.844 | 1.000 | 1.000 |
| FEVER |                       |           |        |
| DeYoung et al. [7] | 21.894 | 0.438 | 0.35 |
| Lei et al. [19] | 138.806 | 0.258 | 0.678 |
| Lehman et al. [18] | 30.882 | 0.584 | 0.508 |
| Bert-To-Bert | 29.127 | 0.904 | 0.811 |
| ExPred (w/o Task Sup.) | 40.742 | 0.868 | 0.816 |
| ExPred | 44.670 | 0.834 | 0.908 |
| HUMAN EXPLANATION | 39.721 | 1.000 | 1.000 |
| MultiRC |                       |           |        |
| DeYoung et al. [7] | 47.699 | 0.337 | 0.352 |
| Lei et al. [19] | 155.696 | 0.182 | 0.365 |
| Lehman et al. [18] | 25.150 | 0.245 | 0.118 |
| Bert-To-Bert | 21.699 | 0.726 | 0.326 |
| ExPred (w/o Task Sup.) | 46.331 | 0.665 | 0.619 |
| ExPred | 55.870 | 0.627 | 0.704 |
| HUMAN EXPLANATION | 49.929 | 1.000 | 1.000 |

Table 2: Performance of soft models, where the metric of Task is Macro F1, the same as in Table 1, Comp represents Comprehensive- ness, the higher the better and Suff is Sufficiency, the lower the better.

| Model Type | Task | AUPRC | Comp↑ | Suff↓ |
|------------|------|-------|-------|-------|
| Movie Reviews |               |       |       |       |
| BERT-LSTM | + Attention | 0.970 | 0.417 | 0.129 | 0.097 |
|            | + Gradient   | 0.970 | 0.385 | 0.142 | 0.112 |
| ExPred-SOFT |              | 0.880 | 0.420 | 0.385 | 0.163 |
| FEVER |                   |       |       |       |       |
| GloVe-LSTM | + Attention | 0.870 | 0.235 | 0.037 | 0.122 |
|            | + Simple Gradient | 0.870 | 0.232 | 0.059 | 0.136 |
| ExPred-SOFT |              | 0.914 | 0.836 | 0.151 | 0.068 |
| MultiRC |                           |       |       |       |       |
| BERT-LSTM | + Attention | 0.655 | 0.244 | 0.036 | 0.052 |
|            | + Simple Gradient | 0.655 | 0.224 | 0.077 | 0.064 |
| ExPred-SOFT |              | 0.726 | 0.695 | 0.157 | 0.031 |
We have essentially one hyperparameter $\lambda$ from Equation 2 that trades-off task and explanation losses during the explanation generation phase. Since our key objective is to strike an effective balance between task performance and explanation accuracy, we validate our model on a metric that is a simple linear combination of task performance (macro F1) and explanation accuracy (Token F1). We present the effect of $\lambda$ on this combined metric in Figure 4.

It is evident from the figures that different datasets show different patterns on the metric mixing both task and explanation performance. However, in general, the general trend is that of a steep increase followed by a steep deterioration leave the sweet point balancing the task and explanation performance. The $\lambda$ corresponding to the combined metric performance is then selected.

The key takeaway from our experiments on different values of $\lambda$ is that we observe (more-or-less) a stable plateau in the range $\lambda \in [1, 50]$ that exhibits low variability performance task and explanation performance. However, the task performance deteriorates rapidly after $\lambda \geq 50$ (or low importance to task-specific loss) indicating that optimizing purely for explanation generation deteriorates task performance.

5.2 Soft Selection Approaches

So far each input word is either a part of an explanation or not. This is categorized as hard-(selection)-model according to DeYoung et al. [7]. It also presents an alternate view to explanations as multi-variable distributions over tokens derived from features, e.g. self-attention values and name it as soft-(selection)-model. ExPred can be cast into a soft selection approach explanation model by constructing probability distributions from $g^\theta(\cdot)$ scores of each word before computing the binary cross-entropy.

To evaluate soft selection, the following metrics are used:

- **AUPRC**, or area under the precision-recall curve is used for the soft selection models. Since soft-annotation for each token is assigned with a ranking score (sometimes probability of being rationale).

- **Comprehensiveness** of a rationale $r_{ij}$ on instance $i$ and class $j$ is defined as comprehensiveness(r) = $\hat{p}_{ij} - \bar{p}_{ij}$, where $\hat{p}_{ij}$ is model’s prediction on the original input, and $\bar{p}_{ij}$ is prediction over the input where the rationale $r_{ij}$ is stripped.

- **Sufficiency** on the other hand is defined as the complement of the comprehensiveness, sufficiency = $\bar{p}_{ij} - \hat{p}_{ij}$, where $\hat{p}_{ij}$ is the predicted probability using only rationale $r_{ij}$.

Table 3 presents the result of ExPred in the soft selection mode. We observe that ExPred-Soft performs consistently well both in terms of task and rationale selection metrics. A higher value of AUPRC indicates that a better choice of a threshold of per token rationale prediction can help in improving explainability. A higher value of comprehensiveness indicates that ExPred-Soft selects the correct rationales that are responsible for accurate task label prediction i.e., task performance drops significantly without these tokens.

The low value of sufficiency also supports the fact i.e., it is an indication that the model can learn the task well only based on those tokens. For Movie Reviews, BERT-LSTM + Attention can identify rationales well (low sufficiency) and the high value of AUPRC indicates that rationales are following human-annotated ones. However, the low value of comprehensiveness reveals that the model can still learn without those rationales. Similar effects were observed in previous work where it was found that attention-based selections are not always rationales [13].

On the other hand, ExPred-Soft ensures that the rationales learned are in accordance with human rationales and the model performance significantly drops without those tokens. It fits with our objective that the models should be interpretable by design. ExPred-Soft performs well both in terms of comprehensiveness and sufficiency for FEVER and MultiRC. For Movie Reviews, ExPred-Soft achieves high comprehensiveness but sufficiency is higher (worse) than the baselines. This suggests that ExPred can retrieve rationale tokens well but those are not sufficient to learn the task, i.e., it fails to capture some rationale tokens. However, it can maintain a balance between task and rationale selection.

5.3 Machine explanations vs Human explanation

From the previous results, it is tempting to conclude that we improve task performance at the expense of being less congruent with human rationales and vice versa. Towards getting a clearer understanding we perform some further analysis to compare explanations generated by our approach vs human rationales. We present
Suicide Kings is a terrible film. Walken aside, there isn't a single appealing cast member...in an amusingly unironic scene...O'fallon is someone whom I'm betting has seen reservoir dogs and the usual suspects too many times...but the central plot itself is a serpentine mess, filled with crosses and double crosses...

Has seen reservoir dogs and the usual suspects too many times...but the central plot itself is a serpentine mess, filled with crosses and double crosses...

Suicide Kings is a terrible film. Walken aside, there isn't a single appealing cast member...in an amusingly unironic scene...O'fallon is someone whom I'm betting has seen reservoir dogs and the usual suspects too many times...but the central plot itself is a serpentine mess, filled with crosses and double crosses...

Suicide Kings is a terrible film. Walken aside, there isn't a single appealing cast member...in an amusingly unironic scene...O'fallon is someone whom I'm betting has seen reservoir dogs and the usual suspects too many times...but the central plot itself is a serpentine mess, filled with crosses and double crosses...
proposes a further human evaluation of the machine-generated explanations. Since our objective in this paper is to generate proper rationales that are sufficient to make predictions, such human evaluation is left for future work.

6 CONCLUSIONS

In this paper we propose a novel yet simple approach ExPred, that uses multi-task learning in the explanation generation phase to provide better task-aware explanations for explain-then-predict models. We find that we substantially outperform existing explain-then-predict approaches by 7% - 47% by explicitly incorporating task-specific supervision during explanation generation. Additionally, we observed that we can also use ExPred in the soft selection setting and observe competitive results. Our main observation is that simple pipeline models like ExPred can indeed strike a good balance between explanation quality and task performance, consistently performing at par or even better than models when given full inputs. This is in contrast to joint models like [19] that find it difficult to incorporate rationales data and are hard to train in general and difficult to maintain.

There are many avenues for future work that are possible. First, end-to-end models outperform ExPred in task performance for Movie Reviews dataset indicating that for some tasks rationale data might be limited or might not be sufficient to deliver better task performance. We would want to scale rationale collection methods and study the impact of the size of the rationale dataset on task performance. We would also want to extend our current pipelined approach to an end-to-end approach. Finally, an important open question that this work prompts is that can extractive explanations be generalized to other Web tasks like search [10, 30] and structured data [8].

Acknowledgement: Funding for this project was in part provided by the European Union’s Horizon 2020 research and innovation program under grant agreement No 832921, and No 871042.

REFERENCES

[1] J. Johnson A. Karpathy and F. Li. 2015. Visualizing and understanding recurrent networks. arXiv preprint arXiv:1506.02078.
[2] J. Bastings, W. Asiz, and I. Tittov. 2019. Interpretable neural predictions with differentiable binary variables. In Proc. ACL, pages 2965–2977.
[3] Rich Caruana. 1997. Multitask learning. Machine learning, 28(1):41–75.
[4] N. Chawla, K. Bowyer, L. Hall, and P. Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16:321–357.
[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proc. NAACL-HLT, pages 4171–4186.
[6] J. DeYoung, S. Jain, N. F. Rajani, E. Lehman, C. Xiong, R. Socher, and B. C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In Proc. ACL, pages 4443–4458.
[7] B. Fetahi, A. Anand, and M. Koutrakis. 2019. Tablenet: An approach for determining fine-grained relations for wikipedia tables. In The World Wide Web Conference, pages 2736–2742.
[8] M. Hermans and B. Schrauwen. 2013. Training and analysing deep recurrent neural networks. In M. Welling C. J. C. Burges, L. Bottou, Z. Ghahramani, and S. M. Roberts, editors, Advances in Neural Information Processing Systems 26, pages 190–198.
[9] H. Holzmann, W. Nejdl, and A. Anand. 2017. Exploring web archives through temporal anchor texts. In Proceedings of the 2017 ACM on Web Science Conference, pages 289–298.
[10] Helge Holzmann, Wolfgang Nejdl, and Avishek Anand. 2016. The dawn of today’s popular domains: A study of the archived german web over 18 years. In 2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL), pages 73–82. IEEE.