Rationales for Sequential Predictions

Anonymous EMNLP submission

Abstract

Sequence models are a critical component of modern NLP systems, but their predictions are difficult to explain. We consider model explanations though *rationales*, subsets of context that can explain individual model predictions. We find sequential rationales by solving a combinatorial optimization: the best rationale is the smallest subset of input tokens that would predict the same output as the full sequence. Enumerating all subsets is intractable, so we propose an efficient greedy algorithm to approximate this objective. The algorithm, which is called greedy rationalization, applies to any model. For this approach to be effective, the model should form compatible conditional distributions when making predictions on incomplete subsets of the context. This condition can be enforced with a short fine-tuning step. We study greedy rationalization on language modeling and machine translation. Compared to existing baselines, greedy rationalization is best at optimizing the sequential objective and provides the most faithful rationales. On a new dataset of annotated sequential rationales, greedy rationales are most similar to human rationales.

1 Introduction

Sequence models are a critical component of tasks ranging from language modeling (Radford et al., 2019) to machine translation (Brown et al., 1993; Vaswani et al., 2017) to summarization (Rush et al., 2015). These tasks are dominated by complex neural networks. While these models produce accurate predictions, their decision making processes are hard to explain. Interpreting a model's prediction is important in a variety of settings: a researcher needs to understand a model to debug it; a doctor using a diagnostic model requires justifications to validate a decision; a company deploying a language model relies on model explanations to detect biases appropriated from training data.

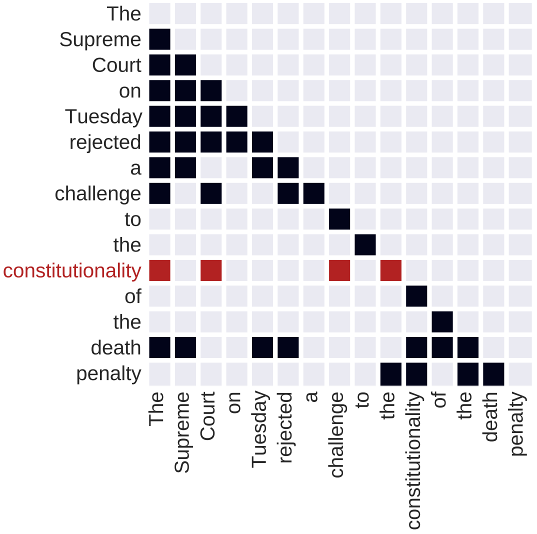


Figure 1. Rationales for sequential prediction on GPT-2. Each row is a predicted word. The darkened cells correspond to the context words found by greedy rationalization. To predict "constitutionality", the model only needs "The", "Court", "challenge", and "the".

Interpretation takes many flavors (Lipton, 2018). We focus on *rationales*, i.e. identifying the most important subset of input tokens that leads to the model's prediction. For example, consider the sentence: "The Supreme Court on Tuesday rejected a challenge to the constitutionality of the death penalty." Suppose we would like to explain the decision of the model to generate "constitutionality." While the model mathematically conditions on all the previous words, only some are critical to its predictions. In this case, the rationale produced by our algorithm includes "the", "challenge", and notably "Court", but not phrases that add no information like "on Tuesday" (Figure 1).

Various rationale methods have been proposed for sequence classification, where each sequence has a single rationale (Lei et al., 2016; Chen et al., 2018; Jain et al., 2020). However, these methods cannot scale to sequence models, where each token in a sequence requires a different rationale.

This work frames the problem of finding sequence rationales as a combinatorial optimization: given a model, the best rationale is the smallest subset of input tokens that would predict the same token as the full sequence. Finding the global optimum in this setting is intractable, so we propose **greedy rationalization**, a greedy algorithm that iteratively builds longer rationales. This approach is efficient for many NLP models such as transformers. Moreover, it does not require access to the inner workings of a model, such as gradients.

Underlying this approach is an assumption that the model forms sensible predictions for incomplete subsets of the input. Although we can pass in incomplete subsets to neural models, there is no guarantee that their predictions on these subsets will be compatible with their predictions on full contexts (Arnold and Press, 1989). We show that compatibility can be learned by conditioning on randomly sampled context subsets while training a model. For large pretrained models like GPT-2 (Radford et al., 2019), fine-tuning is sufficient.

In an empirical study, we compare greedy rationalization to various gradient- and attention-based explanation methods on language modeling and machine translation. Greedy rationalization best optimizes the objective, and its rationales are most faithful to the inner workings of the model. We additionally create a new dataset of annotated rationales based on the Lambada corpus (Paperno et al., 2016). We find that greedy rationales are most similar to human annotations, both on our dataset and on a labeled dataset of translation alignments.

2 Sequential Rationales

Consider a sequence of tokens, $y_{1:T}$, generated by some unknown process $y_{1:T} \sim F$. The goal of sequence modeling is to learn a probabilistic model p_θ that approximates F from samples. Maximum-likelihood estimation is an effective way to train these models, where θ is fit according to

$$\arg\max_{\theta} \mathbb{E}_{y_{1:T} \sim F}[\log p_\theta(y_{1:T})]. \qquad (1)$$

Sequence models are typically factored into conditional distributions:

$$p_\theta(y_{1:T}) = f_\theta(y_1) \prod_{t=2}^{T} f_\theta(y_t | y_{<t}). \qquad (2)$$

Here, f_θ is the specific model parameterizing p_θ, such as a transformer (Vaswani et al., 2017), and is trained to take inputs $y_{<t}$. Going forward, we drop the dependence on θ in the notation.

Word-level explanations are a natural way to interpret a sequence model: which words were instrumental for predicting a particular word? Would the same word have been predicted if some of the words had been missing?

Explanations may be straightforward for simpler models; for example, a bigram Markov model uses only the previously generated word to form predictions. However, the most effective sequence models have been based on neural networks, whose predictions are challenging to interpret (Lipton, 2018).

Motivated by this goal, we consider a sequence $y_{1:T}$ generated by a sequence model p. At each position t, the model takes the inputs in the context $y_{<t}$ and uses them to predict y_t. We are interested in forming *rationales*: subsets of the contexts that can explain the model's prediction of y_t.[1]

What are the properties of a good rationale? Any of the contextual words $y_{<t}$ can contribute to y_t. However, if a model makes the same prediction with only a subset of the context, that subset contains explanatory power on its own. A rationale is *sufficient* if the model would produce the same y_t having seen only the rationale (DeYoung et al., 2020). While rationales consisting of the full context would always be sufficient, they would be ineffective for explaining longer sequences. Intuitively, the smaller the rationale, the easier it is to interpret, so we also prioritize *brevity*.

We combine these desiderata and frame finding rationales as a combinatorial optimization: the best rationale of a word y_t is the smallest subset of inputs that would lead to the same prediction. Each candidate rationale S is an index set, and y_S denotes the subset of tokens indexed by S.[2] Denote by $\mathcal{S} = 2^{[t-1]}$ the set of all possible context subsets. An optimal rationale is given by

$$\arg\min_{S \in \mathcal{S}} |S| \;\; \text{s.t.} \;\; \arg\max_{y'_t} p(y'_t | y_S) = y_t. \qquad (3)$$

The constraint guarantees sufficiency, and the objective targets brevity. Although the objective may have multiple solutions, we only require one.

Optimizing Eq. 3 is hindered by a pair of computational challenges. The first challenge is that

[1] Our paradigm and method extend easily to conditional sequence models, such as those used for machine translation. For full details, refer to Appendix A.

[2] A sequence of tokens can be represented as a set of tuples: "The dog walks" becomes $\{(1 : \text{The}), (2 : \text{dog}), (3 : \text{walks})\}$.

solving this combinatorial objective is intractable; framed as a decision problem, it is NP-hard. We discuss this challenge in Section 3. The second challenge is that evaluating distributions conditioned on incomplete context subsets $p(y'_t|y_S)$ involves an intractable marginalization over missing tokens. For now we assume that $f(y'_t|y_S) \approx p(y'_t|y_S)$; we discuss how to enforce this condition in Section 4.

3 Greedy Rationalization

We propose a simple greedy algorithm, **greedy rationalization**, to approximate the solution to Eq. 3. The algorithm starts with an empty rationale. At each step, it considers adding each possible token, and it selects the one that most increases the probability of y_t. This process is repeated until the rationale is sufficient for predicting y_t.[3] Figure 2 provides an overview.

Here is the algorithm. Begin with a rationale $S^{(0)} = \emptyset$. Denoting by $[t-1] = \{1, \dots, t-1\}$, the first rationale set is

$$S^{(1)} = \underset{k \in [t-1]}{\arg\max}\, p(y_t|y_k). \tag{4}$$

At each step, we iteratively add a single word to the rationale, choosing the one that maximizes the probability of the word y_t:

$$S^{(n+1)} = S^{(n)} \cup \underset{k \in [t-1] \setminus S^{(n)}}{\arg\max} \; p(y_t|y_{S^{(n)} \cup k}). \tag{5}$$

We continue iterating Eq. 5 until $\arg\max_{y'_t} p(y'_t|y_{S^{(n)}}) = y_t$. The procedure will always converge, since in the worst case, $S^{(t-1)}$ contains the full context.

This procedure is simple to implement, and it is black-box: it does not require access to the inner workings of a model, like gradients or attention.

While greedy rationalization can be applied to any model, greedy rationalization is particularly effective for set-based models such as transformers. If we assume the rationale size $m = |S|$ is significantly shorter than the size of the context t, greedy rationalization requires no extra asymptotic complexity beyond the cost of a single evaluation.

For transformers, the complexity of each evaluation $f(y_t|y_{<t})$ is quadratic in the input set $O(t^2)$. Each step of greedy rationalization requires evaluating $f(y_t|y_S)$, but y_S can be significantly smaller than $y_{<t}$. A rationale of size m will require m steps to terminate, resulting in a total complexity of $O(m^3 t)$. As long as $m = O(t^{1/3})$, greedy rationalization can be performed with the same asymptotic complexity as evaluating a transformer on the full input, $O(t^2)$. In Appendix C, we empirically verify the efficiency of greedy rationalization.

4 Model Compatibility

Greedy rationalization requires computing conditional distributions $p(y_t|y_S)$ for arbitrary subsets S. Using an autoregressive model, this calculation requires marginalizing over unseen positions. For example, rationalizing a sequence $y_{1:3}$ requires evaluating the candidate rationale $p(y_3|y_1)$, which marginalizes over the model's predictions:

$$p(y_3|y_1) = \sum_k f(y_3|y_1, y_2 = k) f(y_2 = k|y_1).$$

Given the capacity of modern neural networks, it is tempting to pass in incomplete subsets y_S to f and evaluate this instead as $f(y_t|y_S) \approx p(y_t|y_S)$. However, since f is trained only on complete feature subsets $y_{<t}$, incomplete feature subsets y_S are out-of-distribution (Hooker et al., 2019). Evaluating $f(y_3|y_1)$ may be far from the true conditional $p(y_3|y_1)$. In Figure 4, we show that indeed language models like GPT-2 produce poor predictions on incomplete subsets.

4.1 Fine-tuning for Compatibility

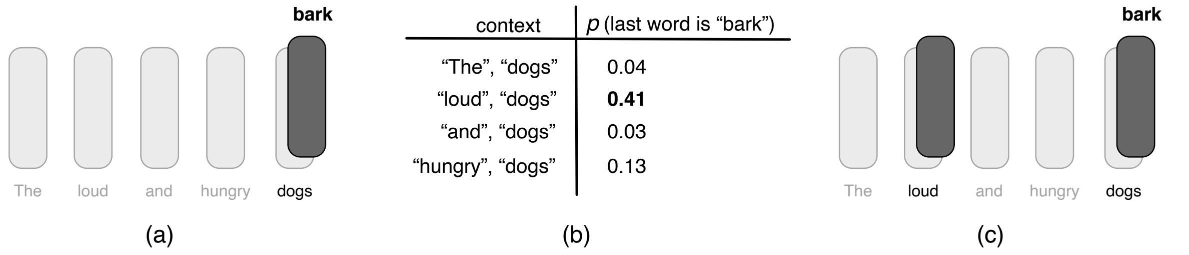
Ideally $f(y_t|y_S)$ approximates $p(y_t|y_S)$, a property known as *compatibility* (Arnold and Press, 1989). Since training with Eq. 1 only evaluates f on complete contexts $y_{<t}$, its behavior on incomplete contexts y_S is unspecified. Instead, compatibility can be obtained by training to maximize

$$\mathbb{E}_{y_{1:T} \sim F} \mathbb{E}_{S \sim \text{Unif}(\mathcal{S})} \left[\textstyle\sum_{t=1}^{T} \log f(y_t|y_{S_{<t}}) \right], \tag{6}$$

where $S \sim \text{Unif}(\mathcal{S})$ indicates sampling word subsets uniformly at random from the power set of all possible word subsets, and $S_{<t}$ denotes the indices in S that are less than t. We approximate Eq. 6 with word dropout.[4] Jethani et al. (2021) show that the optimum of Eq. 6 is the distribution whose conditional distributions are all equal to the ground-truth conditionals.

[3] The greedy approach is motivated by approximations to the set cover problem (Chvatal, 1979). Each set is a single context token, and a rationale is "covered" if it results in generating the true token.

[4] In practice, we combine this objective with standard MLE training to learn compatible distributions while maintaining the performance of the original model. We also skew the word dropout distribution towards sparser rationales since we expect shorter rationales to be more common; see Appendix D.



context	p (last word is "bark")
"The", "dogs"	0.04
"loud", "dogs"	**0.41**
"and", "dogs"	0.03
"hungry", "dogs"	0.13

Figure 2. One step of greedy rationalization. In (a), the rationale so far is a single word, "dogs." In (b), each candidate token is considered and "loud" results in the best probability for "bark." In (c), the token "loud" is added to the rationale. This process repeats until the most likely word is the model prediction.

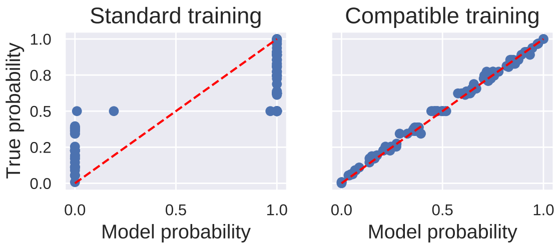


Figure 3. Training with word dropout (right) results in compatible predictions for the majority-class synthetic language. The optimal compatibility is the dashed line.

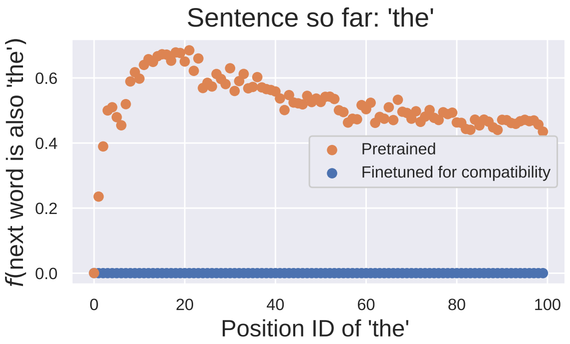


Figure 4. Fine-tuning GPT-2 for compatibility removes pathological repeating on incomplete contexts. For a position t, the vertical axis gives $f(y_{t+1} =$ "the"$|y_t =$ "the").

The intuition for Eq. 6 is straightforward: if the model sees incomplete contexts while training, it can approximate arbitrary incomplete distributions. Since $f(y_t|y_S)$ approximates $F(y_t|y_S)$ and $f(y_t|y_{<t})$ approximates $F(y_t|y_{<t})$, all the conditional distributions are compatible.

4.2 Compatibility Experiments

To demonstrate the impact of training with the compatibility objective in Eq. 6, we consider a synthetic majority-class language over binary strings of 19 tokens. The first 17 are sampled uniformly from $\{0, 1\}$, and the 18th token is always '='. The 19th token is 0 if there are more 0's than 1's in the first 17 tokens, and 1 otherwise.

We train two models: one using the standard objective in Eq. 1, the other using word dropout to optimize Eq. 6. Although both models have the same heldout perplexity on the full context, training with Eq. 6 is required to form compatible predictions on incomplete subsets. In Figure 3, we provide different models f with random subsets S and calculate the model's probability that the last token is 1. A model that has only seen a few tokens should be less confident about the prediction of the final majority class, yet models trained without word dropout ignore this uncertainty.

Models do not need to be trained from scratch with Eq. 6. A model can be pre-trained with Eq. 1, after which it can be fine-tuned for compatibility. As an example, when GPT-2 is not trained with word dropout, it makes insensible predictions for out-of-distribution sequences. For a sequence that contains only the token "the," GPT-2 is trained to give reasonable predictions for $p(y_2|y_1 =$ "the"). But when it has only seen the token "the" somewhere besides the first position of the sequence, the top prediction for the word after "the" is also "the". Of course, following "the" with "the" is not grammatical. Fine-tuning for compatibility alleviates this problem (Figure 4).

Finally, we find that that fine-tuning for compatibility does not hurt the heldout performance of the complete conditional distribution of each fine-tuned model (see Appendix D).

5 Connection to Classification Rationales

In this section, we discuss related approaches developed for classification, and why they cannot scale to sequence models. We also show that the combinatorial rationale objective in Eq. 3 is a global solution to a classification rationale-style objective.

In classification problems, a sequence $x_{1:T}$ is associated with a label y. Rationale methods are commonly used in this setting (Lei et al., 2016; Chen et al., 2018; Yoon et al., 2018; Bastings et al., 2019; Jain et al., 2020; Jethani et al., 2021). The most common approach uses two models: one, a selection model $q(S|x_{1:T})$, provides a distribution over possible rationales; the other, the predictive model $p(y|x_S)$, makes predictions given only samples from the former model. Typically, p and q are optimized jointly to maximize

$$\mathbb{E}_{x,y\sim F}\mathbb{E}_{S\sim q(S|x,y)}[\log p(y|x_S) - \lambda|S|]. \quad (7)$$

Here, F is the ground truth, unknown data distribution, and λ is a regularizing penalty that encourages smaller rationales.

In practice, it is infeasible to adopt this objective for sequence models. Eq. 7 is centered on providing predictive models with only the words in its rationale. In sequential settings, each word requires its own rationale. Thus training with shared word representations would leak information across rationales. Training sequence models without sharing representations is computationally infeasible; it requires $O(T^3)$ computations per sequence for transformer architectures.

Most classification rationale methods treat $q(S|x_{1:T})$ as a probability distribution over all possible rationales. However, the q that maximizes Eq. 7 is deterministic for any p. To see this, note that q does not appear inside the expectation in Eq. 7, so it can place all its mass on a single mode. We provide a formal justification in Appendix B.

Since the optimal selection model q is a point-mass, the optimal rationale can be written as

$$\underset{S\in\mathcal{S}}{\arg\min}\ \lambda|S| - \log p(y|x_S). \quad (8)$$

This optimization is identical to the combinatorial optimization in Eq. 3, albeit with a soft constraint on the rationale's prediction: the true label y is not required to be the maximum of $p(y'|x_S)$. In practice, this soft constraint sometimes results in empty rationales (Jain et al., 2020). Since we view sufficiency as a key component of a good rationale, Eq. 3 imposes a hard constraint on the rationale's prediction.

6 Related Work

Finding rationales is similar to feature selection. While global feature selection has been a well-studied problem in statistics (Guyon and Elisseeff, 2003; Hastie et al., 2009; Bertsimas et al., 2016), instance-wise feature selection — where the goal is selecting features per-example — is a newer research area (Chen et al., 2018). We review local explanation methods used for NLP.

Gradients. Gradient-based saliency methods have long been used as a measure of feature importance in machine learning (Baehrens et al., 2010; Simonyan et al., 2013; Li et al., 2016a). Some variations involve word embeddings (Denil et al., 2014); integrated gradients, to improve sensitivity (Sundararajan et al., 2017); and relevance-propagation to track each input's contribution through the network (Bach et al., 2015; Voita et al., 2021).

But there are drawbacks to using gradient-based methods as explanatory tools. Sundararajan et al. (2017) show that in practice, gradients are *saturated*: they may all be close to zero for a well-fitted function, and thus not reflect importance. Adversarial methods can also distort gradient-based saliences while keeping a model's prediction the same (Ghorbani et al., 2019; Wang et al., 2020). We compare to gradient saliency methods in Section 8.

Attention. Recently, NLP practitioners have focused on using attention weights as explanatory tools. The literature has made a distinction between *faithfulness* and *plausibility*. An explanation is faithful if it accurately depicts how a model makes a decision (Jacovi and Goldberg, 2020); an explanation is plausible if it can be understood and interpreted by humans (Wiegreffe and Pinter, 2019). Practitioners have shown that attention-based explanations are generally not faithful (Jain and Wallace, 2019; Serrano and Smith, 2019), but that they may be plausible (Wiegreffe and Pinter, 2019; Mohankumar et al., 2020; Vashishth et al., 2019). Others show that attention weights should not be interpreted as belonging to single tokens since they mix information across tokens (Brunner et al., 2019; Kobayashi et al., 2020). Bastings and Filippova (2020) argue that general input saliency measures, such as gradients, are better suited for explainability than attention. We compare to attention-based methods in Section 8.

Local post-hoc interpretability. Another class of methods provides local interpretability for pretrained models. These approaches aim to explain a model's behavior for a single example or for a small subset of inputs. LIME (Ribeiro et al., 2016) trains an interpretable model that locally approximates the pretrained model. Alvarez-Melis and Jaakkola (2017) learn a causal relationship between perturbed inputs and their model outputs. These methods impose no constraints on the pretrained model. However, they are expensive – they require training separate models for each input region. In contrast, the method proposed here, greedy rationalization, can efficiently explain many predictions.

Input perturbation. Practitioners have also measured the importance of inputs by perturbing them (Zeiler and Fergus, 2014; Kádár et al., 2017). Occlusion methods (Li et al., 2016b) replace an input with a baseline (e.g. zeros), while omission methods (Kádár et al., 2017) remove words entirely. Li et al. (2016b) propose a reinforcement learning method that aims to find the minimum number of occluded words that would change a model's prediction. Feng et al. (2018) use gradients to remove unimportant words to see how long it takes for the model's prediction to change. They find that the remaining words are nonsensical and do not comport with other saliency methods. Others have shown that input perturbation performs worse than other saliency methods in practice (Poerner et al., 2018). These methods have mostly focused on subtractive techniques. For this reason, they are inefficient and do not aim to form sufficient explanations. In contrast, greedy rationalization efficiently builds up sufficient explanations.

7 Experimental Setup

There are two goals in our empirical studies. The first is to compare the ability of greedy rationalization to other approaches for optimizing the combinatorial objective in Eq. 3. The second is to assess the quality of produced rationales.

We measure the quality of rationales using two criteria: faithfulness and plausibility. An explanation is faithful if it accurately depicts how a model makes a decision (Jacovi and Goldberg, 2020); an explanation is plausible if it can be understood and interpreted by humans (Wiegreffe and Pinter, 2019). Although sufficiency is a standard way to measure faithfulness (DeYoung et al., 2020), all the rationales that satisfy the constraint of Eq. 3 are sufficient by definition. To measure plausibility, we compare rationales to human annotations. Since there do not exist language modeling datasets with human rationales, we collected annotations based on Lambada (Paperno et al., 2016). The data is available as part of this paper.

We compare greedy rationalization to a variety of gradient- and attention-based baselines (see Section 6). To form baseline sequential rationales, we add words by the order prescribed by each approach, stopping when the model prediction is sufficient. The baselines are: l_2 gradient norms of embeddings (Li et al., 2016a), embedding gradients multiplied by the embeddings (Denil et al., 2014), integrated gradients (Sundararajan et al., 2017), attention rollout (Abnar and Zuidema, 2020), the last-layer transformer attention weights averaged-across heads, and all transformer attentions averaged across all layers and heads (Jain et al., 2020).

To compare rationale sets produced by each method to those annotated by humans, we use the set-similarity metrics described in DeYoung et al. (2020): the intersection-over-union (IOU) of each rationale and the human rationale, along with the token-level F1, treating tokens as binary predictions (either in the human rationale or out of it).

We use transformer-based models for all of the experiments.[5] We will release our fine-tuned GPT-2 model on Hugging Face (Wolf et al., 2019). For model and fine-tuning details, refer to Appendix D.

8 Results and Discussion

The experiments test sequential rationales for language modeling and machine translation. Appendix E contains full details for each experiment.

8.1 Language Modeling

Long-Range Agreement. The first study tests whether rationales for language models can capture long-range agreement. We create a template dataset using the analogies from Mikolov et al. (2013). This dataset contains word pairs that contain either a semantic or syntactic relationship. For each type of relationship, we use a predefined template. It prompts a language model to complete the word pair after it has seen the first word.

For example, one of the fifteen categories is countries and their capitals. We can prompt a language model to generate the capital by first mentioning a country and then alluding to its capital. To test long-range agreement, we also include a distractor sentence that contains no pertinent information about the word pair. For example, our

[5] We fine-tune each model for compatibility using a single GPU. That we can fine-tune GPT-2 Large (Radford et al., 2019) to learn compatible conditional distributions on a single GPU suggests that most practitioners will be able to train compatible models using a reasonable amount of computation.

	Length	Ratio	Ante	No D
Grad norms	22.5	4.1	**1.0**	0.06
Grad x emb	38.0	7.4	0.99	0.01
Integrated grads	28.1	5.2	0.99	0.00
Attention rollout	36.9	7.1	**1.0**	0.12
Last attention	16.7	2.9	0.99	0.13
All attentions	14.5	2.6	**1.0**	0.02
Greedy	**7.1**	**1.2**	**1.0**	**0.43**

Table 1. Language modeling faithfulness on long-range agreement with templated analogies. "Ratio" refers to the approximation ratio of each method's rationale length to the exhaustive search minimum. "Ante" refers to the percent of rationales that contain the true antecedent. "No D" refers to the percent of rationales that do not contain any tokens from the distractor.

template for this category is,

> When my flight landed in **Japan**, I converted my currency and slowly fell asleep. (I had a terrifying dream about my grandmother, but that's a story for another time). I was staying in the capital, ______________

Here, the parenthetical clause is a distractor sentence, since it contains no relevant information about predicting the capital of Japan. The correct capital, "Tokyo," is predicted by GPT-2 both with and without the distractor. We use this template for all of the examples in the country capital category, swapping the antecedent "Japan" for each country provided in Mikolov et al. (2013).

We feed the prompts to GPT-2, which completes each analogy. To measure faithfulness, we calculate the percent of rationales that contain the true antecedent, and the percent of rationales that do not contain any words in the distractor. We only use examples where the prediction is the same both with and without the distractor. We also perform exhaustive rationale search on the objective in Eq. 3. This search is highly inefficient, so we only complete it for 40 examples. To measure the approximation ratio, we divide the size of the rationale found by each method by the exhaustive rationale size.

Table 1 contains the results on the compatible model.[6] Although all methods contain the true antecedents in their rationales, greedy rationalization has by far the least distractors in its rationales. The rationales are also universally shorter for greedy rationalization, and closer to the optimal rationales, justifying our greedy assumption.

	Length	IOU	F1
Gradient norms	52.9	0.13	0.21
Gradient x embedding	64.8	0.11	0.19
Integrated gradients	59.1	0.11	0.19
Attention rollout	73.5	0.09	0.17
Last attention layer	43.2	0.17	0.27
All attention layers	35.8	0.24	0.33
Greedy	**14.1**	**0.27**	**0.37**

Table 2. Language modeling plausibility on rationale-annotated Lambada.

Annotated Rationales. To test the plausibility of rationales for language models, we collect a dataset of human annotations. We base the collection on Lambada (Paperno et al., 2016). Lambada is constructed so that humans need to use both local and global context to reliably predict a missing word. By its construction it is guaranteed to have non-trivial rationales.

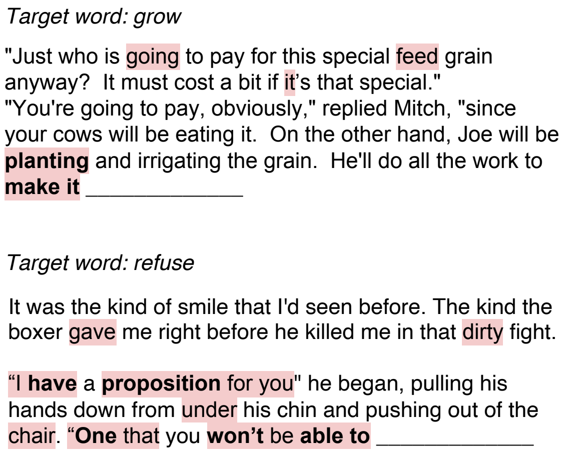
Our goal is to collect rationales that are both minimal and sufficient for humans. We run an annotation procedure with two roles: a selector and a predictor. First, the selector sees the full passage and ranks the words in order of how informative they are for predicting the final word. Next, the predictor sees one word at a time chosen by the selector, and is asked to predict the final word of the passage. The words the predictor saw before guessing the correct word form a human rationale. This rationale selection method is inspired by Rissanen Data Analysis (Rissanen, 1978; Perez et al., 2021), which uses a minimum description length metric to estimate feature importances. We rely on human annotators to estimate information gains.

Since it could be trivial for humans to predict the final word if it also appears in the context, we only include examples that do not repeat a word. We collect annotations for 107 examples, which we also release publicly. We compare the rationales produced by each method to the annotated rationales. In the analysis, we only include the 62 examples that GPT-2 predicts correctly.

Table 2 shows that the greedy rationales are most similar to the human-annotated rationales. Greedy rationalization is also the most effective at minimizing the combinatorial objective in Eq. 3, as its rationales are by far the shortest. Figure 5 contains examples of rationales for this dataset.

It is worth noting that the top few words added by the baselines are quite relevant; after 5 tokens, the "All attention layers" baseline has a better F1 and IOU than greedy rationalization. However,

[6] To show that fine-tuning GPT-2 for compatibility is not hurting the baselines, we also perform the baseline methods on a pretrained GPT-2 without fine-tuning; see Appendix E.



Target word: grow

"Just who is going to pay for this special feed grain anyway? It must cost a bit if it's that special."
"You're going to pay, obviously," replied Mitch, "since your cows will be eating it. On the other hand, Joe will be **planting** and irrigating the grain. He'll do all the work to **make it** _____________

Target word: refuse

It was the kind of smile that I'd seen before. The kind the boxer gave me right before he killed me in that dirty fight.

"I **have** a **proposition** for you" he began, pulling his hands down from under his chin and pushing out of the chair. "**One** that you **won't** be **able to** _____________

Figure 5. Examples from our annotated Lambada dataset. Highlighted text denotes greedy rationales, and **bolded text** denotes human-annotated rationales.

	Mean Crossovers		Crossover Rate	
	Source	Target	Source	Target
Grad norms	0.41	0.50	**0.06**	0.07
Grad x emb	6.22	5.63	0.42	0.42
Integrated grads	1.93	1.53	0.22	0.12
Last attention	0.56	2.49	0.08	0.24
All attentions	0.60	0.83	0.08	0.11
Greedy	**0.11**	**0.16**	0.08	**0.03**

Table 3. Translation faithfulness with distractors. "Mean crossovers" refers to the average number of crossovers per rationale, and "Crossover rate" refers to the fraction of rationales that contain at least one.

	Length	AER ↓	IOU	F1	Top1
Grad norms	10.3	0.82	0.31	0.16	0.62
Grad x emb	13.0	0.89	0.16	0.12	0.40
Integrated grads	11.0	0.84	0.27	0.14	0.45
Last attention	10.7	0.83	0.28	0.15	0.59
All attentions	10.6	0.82	0.32	0.15	**0.65**
Greedy	**5.0**	**0.77**	**0.41**	**0.23**	0.64

Table 4. Translation plausibility with annotated alignments. The first four columns correspond to using the full rationale found by each method; the last column "Top1" refers to the accuracy of the first token added by each method. AER refers to alignment error rate.

the baselines struggle to form sufficient rationales, which hurts their overall performance.

8.2 Machine Translation

Distractors. To measure faithfulness, we take a transformer trained on IWSLT14 De-En (and fine-tuned for compatibility), and generate translations for 1000 source sequences. We then randomly concatenate each source sequence with a distractor (before or after). We know that each target sequence is generated from the original source. Thus, we can evaluate rationales by penalizing them for "crossing over" to the distractor.

Table 3 contains the results. Greedy rationalization has by far the fewest average number of crossovers per rationale. Although the percent of source rationales that cross over is slightly higher than the percent using gradient norms, the percentage on the target side is superior.

Annotated Alignments. To test plausibility, we compare the rationales to human-labeled word alignments. Using a dataset containing 500 annotated alignments for German-English translation,[7] we compute rationales for each method using the ground truth targets. We measure similarity to the labeled rationales by computing alignment error rate (AER) (Och and Ney, 2000), along with computing the IOU and F1 between sets. To separate the requirement that the rationale be sufficient from each method's global ordering of tokens, we also compare top-1 accuracies, which measure whether the top token identified by each baseline is present in the labeled alignment set.

Table 4 contains the results. The rationales learned by greedy rationalization are more similar to human-labeled alignments than those provided by gradient and attention methods. Many methods have similar top-1 accuracies — indeed, the best top-1 accuracy comes from averaging all attention layers. This reinforces the notion that although the baselines may be able to capture first-order information, they struggle to form sufficient rationales.

9 Conclusion

We proposed an optimization-based algorithm for rationalizing sequence predictions. Although exact optimization is intractable, we developed a greedy approach that efficiently finds good rationales. Moreover, we showed that models can be fine-tuned to form compatible distributions, thereby circumventing an intractable marginalization step. In experiments, we showed that the greedy algorithm is effective at optimization, and that its rationales are more faithful and plausible than those of gradient- and attention-based methods. We hope that our research, along with the release of an annotated dataset of sequence rationales, catalyzes further research into this area.

[7] https://www-i6.informatik.rwth-aachen.de/goldAlignment/

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Association for Computational Linguistics*.

David Alvarez-Melis and Tommi S Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Association for Computational Linguistics*.

Barry C Arnold and S James Press. 1989. Compatible conditional distributions. *Journal of the American Statistical Association*, 84(405):152–156.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Association for Computational Linguistics*.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *ACL Workshop on BlackboxNLP*.

Dimitris Bertsimas, Angela King, Rahul Mazumder, et al. 2016. Best subset selection via a modern optimization lens. *Annals of Statistics*, 44(2):813–852.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. In *Association for Computational Linguistics*.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2019. On identifiability in transformers. In *International Conference on Learning Representations*.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT evaluation campaign. In *International Workshop on Spoken Language Translation*.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*.

Vasek Chvatal. 1979. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235.

Misha Denil, Alban Demiraj, and Nando De Freitas. 2014. Extraction of salient sentences from labelled documents. In *International Conference on Learning Representations*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Association for Computational Linguistics*.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Association for Computational Linguistics*.

Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of neural networks is fragile. In *Association for the Advancement of Artificial Intelligences*.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The elements of statistical learning: Data mining, inference, and prediction*. Springer Science & Business Media.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2019. A benchmark for interpretability methods in deep neural networks. *Neural Information Processing Systems*.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Association for Computational Linguistics*.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *North American Chapter of the Association for Computational Linguistics*.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C Wallace. 2020. Learning to faithfully rationalize by construction. In *Association for Computational Linguistics*.

Neil Jethani, Mukund Sudarshan, Yindalon Aphinyanaphongs, and Rajesh Ranganath. 2021. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In *Artificial Intelligence and Statistics*.

Akos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of linguistic form and function in recurrent neural networks. In *Association for Computational Linguistics*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Association for Computational Linguistics*.

T. Lei, R. Barzilay, and T. Jaakkola. 2016. Rationalizing neural predictions. In *Association for Computational Linguistics*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in NLP. In *Association for Computational Linguistics*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Zachary C Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. In *Queue*, volume 16, pages 31–57. ACM New York, NY, USA.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Workshop Track at ICLR*.

Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2020. Towards transparent and explainable attention models. In *Association for Computational Linguistics*.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Association for computational linguistics*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. In *Association for Computational Linguistics*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Association for Computational Linguistics*.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. Rissanen data analysis: Examining dataset characteristics via description length. *arXiv preprint arXiv:2103.03872*.

Nina Poerner, Benjamin Roth, and Hinrich Schütze. 2018. Evaluating neural network explanation methods using hybrid documents and morphological agreement. In *Association for Computational Linguistics*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *Special Interest Group on Knowledge Discovery and Data*.

Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica*, 14(5):465–471.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Association for Computational Linguistics*.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? In *Association for Computational Linguistics*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across NLP tasks. *arXiv preprint arXiv:1909.11218*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021. Analyzing the source and target contributions to predictions in neural machine translation. In *Association for Computational Linguistics*.

Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. 2020. Gradient-based analysis of NLP models is manipulable. In *Empirical Methods in Natural Language Processing*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Empirical Methods in Natural Language Processing*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. In *Empirical Methods in Natural Language Processing*.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. 2018. INVASE: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*.