A causal framework for explaining the predictions of black-box sequence-to-sequence models

David Alvarez-Melis and **Tommi S. Jaakkola**
CSAIL, MIT
{davidam, tommi}@csail.mit.edu

Abstract

We interpret the predictions of any black-box structured input-structured output model around a specific input-output pair. Our method returns an "explanation" consisting of groups of input-output tokens that are causally related. These dependencies are inferred by querying the black-box model with perturbed inputs, generating a graph over tokens from the responses, and solving a partitioning problem to select the most relevant components. We focus the general approach on sequence-to-sequence problems, adopting a variational autoencoder to yield meaningful input perturbations. We test our method across several NLP sequence generation tasks.

1 Introduction

Interpretability is often the first casualty when adopting complex predictors. This is particularly true for structured prediction methods at the core of many natural language processing tasks such as machine translation (MT). For example, deep learning models for NLP involve a large number of parameters and complex architectures, making them practically black-box systems. While such systems achieve state-of-the-art results in MT (Bahdanau et al., 2014), summarization (Rush et al., 2015) and speech recognition (Chan et al., 2015), they remain largely uninterpretable, although attention mechanisms (Bahdanau et al., 2014) can shed some light on how they operate.

Stronger forms of interpretability could offer several advantages, from trust in model predictions, error analysis, to model refinement. For example, critical medical decisions are increasingly being assisted by complex predictions that should lend themselves to easy verification by human experts. Without understanding how inputs get mapped to the outputs, it is also challenging to diagnose the source of potential errors. A slightly less obvious application concerns model improvement (Ribeiro et al., 2016) where interpretability can be used to detect biases in the methods.

Interpretability has been approached primarily from two main angles: *model interpretability*, i.e., making the architecture itself interpretable, and *prediction interpretability*, i.e., explaining particular predictions of the model (cf. (Lei et al., 2016)). Requiring the model itself to be transparent is often too restrictive and challenging to achieve. Indeed, prediction interpretability can be more easily sought *a posteriori* for black-box systems including neural networks.

In this work, we propose a novel approach to prediction interpretability with only oracle access to the model generating the prediction. Following (Ribeiro et al., 2016), we turn the local behavior of the model around the given input into an interpretable representation of its operation. In contrast to previous approaches, we consider structured prediction where both inputs and outputs are combinatorial objects, and our explanation consists of a summary of operation rather than a simpler prediction method.

Our method returns an "explanation" consisting of sets of input and output tokens that are causally related under the black-box model. Causal dependencies arise from analyzing perturbed versions of inputs that are passed through the black-

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 412–421
Copenhagen, Denmark, September 7–11, 2017.

box model. Although such perturbations might be available in limited cases, we generate them automatically. For sentences, we adopt a variational autoencoder to produce semantically related sentence variations. The resulting inferred causal dependencies (interval estimates) form a dense bipartite graph over tokens from which explanations can be derived as robust min-cut k-partitions.

We demonstrate quantitatively that our method can recover known dependencies. As a starting point, we show that a grapheme-to-phoneme dictionary can be largely recovered if given to the method as a black-box model. We then show that the explanations provided by our method closely resemble the attention scores used by a neural machine translation system. Moreover, we illustrate how our summaries can be used to gain insights and detect biases in translation systems. Our main contributions are:

- We propose a general framework for explaining structured black-box models
- For sequential data, we propose a variational autoencoder for controlled generation of input perturbations required for causal analysis
- We evaluate the explanations produced by our framework on various sequence-to-sequence prediction tasks, showing they can recover known associations and provide insights into the workings of complex systems.

2 Related Work

There is a wide body of work spanning various fields centered around the notion of "interpretability". This term, however, is underdetermined, so the goals, methods and formalisms of these approaches are often non-overlapping (Lipton, 2016). In the context of machine learning, perhaps the most visible line of work on interpretability focuses on medical applications (Caruana et al., 2015), where trust can be a decisive factor on whether a model is used or not. With the ever-growing success and popularity of deep learning methods for image processing, recent work has addressed interpretability in this setting, usually requiring access to the method's activations and gradients (Selvaraju et al., 2016), or directly modeling how influence propagates (Bach et al., 2015). For a broad overview of interpretability in machine learning, we refer the reader to the recent survey by Doshi-Velez and Kim (2017).

Most similar to this work are the approaches of Lei et al. (2016) and Ribeiro et al. (2016). The former proposes a model that justifies its predictions in terms of fragments of the input. This approach formulates explanation generation as part of the learning problem, and, as most previous work, only deals with the case where predictions are scalar or categorical. On the other hand, Ribeiro et al. (2016) propose a framework for explaining the predictions of black-box classifiers by means of locally-faithful interpretable models. They focus on sparse linear models as explanations, and rely on local perturbations of the instance to explain. Their model assumes the input directly admits a fixed size interpretable representation in euclidean space, so their framework operates directly on this vector-valued representation.

Our method differs from—and can be thought of as generalizing—these approaches in two fundamental aspects. First, our framework considers both inputs and outputs to be structured objects thus extending beyond the classification setting. This requires rethinking the notion of explanation to adapt it to variable-size combinatorial objects. Second, while our approach shares the locality and model-agnostic view of Ribeiro et al. (2016), generating perturbed versions of structured objects is a challenging task by itself. We propose a solution to this problem in the case of sequence-to-sequence learning.

3 Interpreting structured prediction

Explaining predictions in the structured input-structured output setting poses various challenges. As opposed to scalar or categorical prediction, structured predictions vary in size and complexity. Thus, one must decide not only how to explain the prediction, but also what parts of it to explain. Intuitively, the "size" of an explanation should grow with the size of the input and output. A good explanation would ideally also decompose into *cognitive chunks* (Doshi-Velez and Kim, 2017): basic units of explanation which are a priori bounded in size. Thus, we seek a framework that naturally

decomposes an explanation into (potentially several) *explaining components*, each of which justifies, from the perspective of the black-box model, parts of the output relative to the parts of the input.

Formally, suppose we have a black-box model $F : \mathcal{X} \rightarrow \mathcal{Y}$ that maps a structured input $\mathbf{x} \in \mathcal{X}$ to a structured output $\mathbf{y} \in \mathcal{Y}$. We make no assumptions on the spaces $\mathcal{X}, \mathcal{Y}$, except that their elements admit a feature-set representation $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$. Thus, $\mathbf{x}$ and $\mathbf{y}$ can be sequences, graphs or images. We refer to the elements x_i and y_j as units or "tokens" due to our motivating application of sentences, though everything in this work holds for other combinatorial objects.

For a given input output pair $(\mathbf{x}, \mathbf{y})$, we are interested in obtaining an *explanation* of $\mathbf{y}$ in terms of $\mathbf{x}$. Following (Ribeiro et al., 2016), we seek explanations via *interpretable representations* that are both i) *locally faithful*, in the sense that they approximate how the model behaves in the vicinity of $\mathbf{x}$, and ii) *model agnostic*, that is, that do not require any knowledge of F. For example, we would like to identify whether token x_i is a likely cause for the occurrence of y_j in the output when the input context is $\mathbf{x}$. Our assumption is that we can summarize the behavior of F around $\mathbf{x}$ in terms of a weighted bipartite graph $G = (V_x \cup V_y, E)$, where the nodes V_x and V_y correspond to the elements in $\mathbf{x}$ and $\mathbf{y}$, respectively, and the weight of each edge E_{ij} corresponds to the influence of the occurrence of token x_i on the appearance of y_j. The bipartite graph representation suggests naturally that the explanation be given in terms of explaining components. We can formalize these components as subgraphs $G^k = (V_x^k \cup V_y^k, E^k)$, where the elements in V_x^k are likely causes for the elements in V_y^k. Thus, we define an explanation of $\mathbf{y}$ as a collection of such components: $E_{x \rightarrow y} = \{G^1, \ldots, G^k\}$.

Our approach formalizes this framework through a pipeline (sketched in Figure 1) consisting of three main components, described in detail in the following section: a perturbation model for exercising F locally, a causal inference model for inferring associations between inputs and predictions, and a selection step for partitioning and selecting the most relevant sets of associations. We refer to this framework as a *structured-output causal rationalizer* (SOCRAT).

A note on alignment models When the inputs and outputs are sequences such as sentences, one might envision using an alignment model, such as those used in MT, to provide an explanation. This differs from our approach in several respects. Specifically, we focus on explaining the behavior of the "black box" mapping F only locally, around the current input context, not globally. Any global alignment model would require access to substantial parallel data to train and would have varying coverage of the local context around the specific example of interest. Any global model would likely also suffer from misspecification in relation to F. A more related approach to ours would be an alignment model trained locally based on the same perturbed sentences and associated outputs that we generate.

4 Building blocks

4.1 Perturbation Model

The first step in our approach consists of obtaining *perturbed* versions of the input: semantically similar to the original but with potential changes in elements and their order. This is a major challenge with any structured inputs. We propose to do this using a variational autoencoder (VAE) (Kingma and Welling, 2014; Rezende et al., 2014). VAEs have been successfully used with fixed dimensional inputs such as images (Rezende and Mohamed, 2015; Sønderby et al., 2016) and recently also adapted to generating sentences from continuous representations (Bowman et al., 2016). The goal is to introduce the perturbation in the continuous latent representation rather than directly on the structured inputs.

A VAE is composed of a probabilistic encoder $\textsc{Enc} : \mathcal{X} \rightarrow \mathbb{R}^d$ and a decoder $\textsc{Dec} : \mathbb{R}^d \rightarrow \mathcal{X}$. The encoder defines a distribution over latent codes $q(\mathbf{z}|\mathbf{x})$, typically by means of a two-step procedure that first maps $\mathbf{x} \mapsto (\boldsymbol{\mu}, \boldsymbol{\sigma})$ and then samples $\mathbf{z}$ from a gaussian distribution with these parameters. We can leverage this stochasticity to obtain perturbed versions of the input

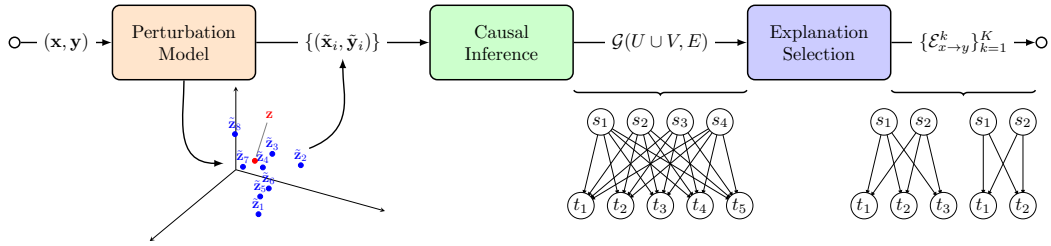


Figure 1: A schematic representation of the proposed prediction interpretability method.

by sampling repeatedly from this distribution, and then mapping these back to the original space using the decoder. The training regime for the VAE ensures approximately that a small perturbation of the hidden representation maintains similar semantic content while introducing small changes in the decoded surface form. We emphasize that the approach would likely fail with an ordinary autoencoder where small changes in the latent representation can result in large changes in the decoded output. In practice, we ensure diversity of perturbations by scaling the variance term $\boldsymbol{\sigma}$ and sampling points $\tilde{\mathbf{z}}$ and different resolutions. We provide further details of this procedure in the supplement. Naturally, we can train this perturbation model in advance on (unlabeled) data from the input domain $\mathcal{X}$, and then use it as a subroutine in our method. After this process is complete, we have N pairs of perturbed input-output pairs: $\{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^N$ which exercise the mapping F around semantically similar inputs.

4.2 Causal model

The second step consists of using the perturbed input-output pairs $\{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^N$ to infer causal dependencies between the original input and output tokens. A naive approach would consider 2x2 contingency tables representing presence/absence of input/output tokens together with a test statistic for assessing their dependence. Instead, we incorporate *all* input tokens simultaneously to predict the occurrence of a single output token via logistic regression. The quality of these dependency estimators will depend on the frequency with which each input and output token occurs in the perturbations. Thus, we are interested in obtaining uncertainty estimates for these predictions, which can be naturally done with a Bayesian approach to logistic regression. Let $\phi_{\mathbf{x}}(\tilde{\mathbf{x}}) \in \{0,1\}^{|\mathbf{x}|}$ be a binary vector encoding the presence of the original tokens $x_1, \ldots, x_n$ from $\mathbf{x}$ in the perturbed version $\tilde{\mathbf{x}}$. For each target token $y_j \in \mathbf{y}$, we estimate a model:

$$P(y_j \in \tilde{\mathbf{y}} \mid \tilde{\mathbf{x}}) = \sigma(\boldsymbol{\theta}_j^T \phi_{\mathbf{x}}(\tilde{\mathbf{x}})) \quad (1)$$

where $\sigma(z) = (1 + \exp(-z))^{-1}$. We use a Gaussian approximation for the logarithm of the logistic function together with the prior $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}_0, \mathbf{H}_0^{-1})$ (Murphy, 2012). Since in our case all tokens are guaranteed to occur at least once (we include the original example pair as part of the set), we use $\boldsymbol{\theta}_0 = \alpha\mathbf{1}, \mathbf{H}_0 = \beta\mathbf{I}$, with $\alpha, \beta > 0$. Upon completion of this step, we have dependency coefficients between all original input and output tokens $\{\theta_{ij}\}$, along with their uncertainty estimates.

4.3 Explanation Selection

The last step in our interpretability framework consists of selecting a set explanations for $(\mathbf{x}, \mathbf{y})$. The steps so far yield a dense bipartite graph between the input and output tokens. Unless $|\mathbf{x}|$ and $|\mathbf{y}|$ are small, this graph itself may not be sufficiently interpretable. We are interested in selecting *relevant* components of this dependency graph, i.e., partition the vertex set of $\mathcal{G}$ into disjoint subsets so as to minimize the weight of omitted edges (i.e. the k-cut value of the partition).

Graph partitioning is a well studied NP-complete problem (Garey et al., 1976). The usual setting assumes deterministic edge weights, but in our case we are interested in incorporating the uncertainty of the dependency estimates—resulting from their finite sample estimation—into the partitioning problem. For this, we rely on the approach of Fan et al. (2012) designed for interval estimates of edge weights. At a high level, this is a robust optimization formulation which seeks to minimize worst case cut values, and can be cast as a Mixed Integer Programming (MIP) problem. Specifically, for a bipartite graph $G = (U, V, E)$

Algorithm 1 Structured-output causal rationalizer

procedure SOCRAT($\mathbf{x}, \mathbf{y}, F$)
  $(\boldsymbol{\mu}, \boldsymbol{\sigma}) \leftarrow$ ENCODE($\mathbf{x}$)
  for $i = 1$ **to** N **do**
    $\tilde{\mathbf{z}}_i \leftarrow$ SAMPLE($\boldsymbol{\mu}, \boldsymbol{\sigma}$)
    $\tilde{\mathbf{x}}_i \leftarrow$ DECODE($\tilde{\mathbf{z}}_i$)
    $\tilde{\mathbf{y}}_i \leftarrow F(\tilde{\mathbf{x}}_i)$
  end for } Perturbation Model.
  $G \leftarrow$ CAUSAL($\mathbf{x}, \mathbf{y}, \{\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i\}_{i=1}^N$)
  $E_{x \mapsto y} \leftarrow$ BIPARTITION(G)
  $E_{x \mapsto y} \leftarrow$ SORT($E_{x \mapsto y}$) ▷ By cut capacity
  return $E_{x \mapsto y}$
end procedure

with edge weights given as uncertainty intervals $\theta_{ij} \pm \hat{\theta}_{ij}$, the partitioning problem is given by

$$\min_{(x^u_{ik}, x^v_{jk}, y_{ij}) \in Y} \sum_{i=1}^{n} \sum_{j=1}^{m} \theta_{ij} y_{ij} + \max_{\substack{S: S \subseteq V, |S| \leq \Gamma \\ (i_t, j_t) \in V \setminus S}} \sum_{(i,j) \in S} \hat{\theta}_{ij} y_{ij} + (\Gamma - \lfloor \Gamma \rfloor) \hat{\theta}_{i_t, j_t} y_{i_t, j_t} \tag{2}$$

where x^u_{ik}, x^v_{jk} are binary variables indicating subset belonging for elements of U and V respectively, y_{ij} are binary auxiliary variables indicating whether i and j are in different partitions, and Y is a set of constraints that ensure the K-partition is valid. Γ is a parameter in $[0, |V|]$ which adjusts the robustness of the partition (the number of deviations from the mean edge values). See the supplement for further explanation of this objective.

If $|\mathbf{x}|$ and $|\mathbf{y}|$ are small, the number of clusters K will also be small, so we can simply return all the partitions (i.e. the *explanation chunks*) $E^k_{x \to y} := (V^k_x \cup V^k_y)$. However, when K is large, one might wish to entertain only the κ most relevant explanations. The graph partitioning framework provides us with a natural way to score the importance of each chunk. Intuitively, subgraphs that have few high-valued edges connecting them to other parts of the graph (i.e. low *cut-capacity*) can be thought of as *self-contained* explanations, and thus more relevant for interpretability. We can therefore define the importance score an atom as:

$$\text{importance}(E^k_{x \to y}) := - \sum_{(i,j) \in X_k} \theta_{ij} \tag{3}$$

where X_k is the cut-set implied by $E^k_{x \to y}$:

$$X_k = \{(i,j) \in E \mid i \in E^k_{x \to y},\ j \in V \setminus E^k_{x \to y}\}$$

The full interpretability method is succinctly expressed in Algorithm 1.

5 Experimental Framework

5.1 Training and optimization

For the experiments involving sentence inputs, we train in advance the VAE described in Section 4.1. We use symmetric encoder-decoders consisting of recurrent neural networks with an intermediate variational layer. In our case, however, we use L stacked RNN's on both sides, and a stacked variational layer. Training variational autoencoders for text is notoriously hard. In addition to dropout and KLD annealing (Bowman et al., 2016), we found that slowly scaling the variance sampled from the normal distribution from 0 to 1 made training much more stable.

For the partitioning step we compare the robust formulation described above with two classical approaches to bipartite graph partitioning which do not take uncertainty into account: the coclustering method of Dhillon (2001) and the biclustering method of Kluger et al. (2003). For these two, we use off-the-shelf implementations,[1] while we solve the MIP problem version of (2) with the optimization library `gurobi`.[2]

5.2 Recovering simple mappings

Before using our interpretability framework in real tasks where quantitative evaluation of explanations is challenging, we test it in a simplified setting where the "black-box" is simple and fully known. A reasonable minimum expectation on our method is that it should be able to infer many of these simple dependencies. For this purpose, we use the CMU Dictionary of word pronunciations,[3] which is based on the ARPAbet symbol set and consists of about 130K word-to-phoneme pairs. Phonemes are expressed as tokens of 1 to 3 characters. An example entry in this dictionary is the pair *vowels* ↦ `V AW1 AH0 L Z`. Though the mapping is simple, it is not one-to-one (a group of characters can correspond to a single phoneme) nor deterministic (the same character can map to different phonemes depending on the context). Thus, it provides a reasonable testbed

[1] http://scikit-learn.org/stable/modules/biclustering.html
[2] http://www.gurobi.com/
[3] www.speech.cs.cmu.edu/cgi-bin/cmudict

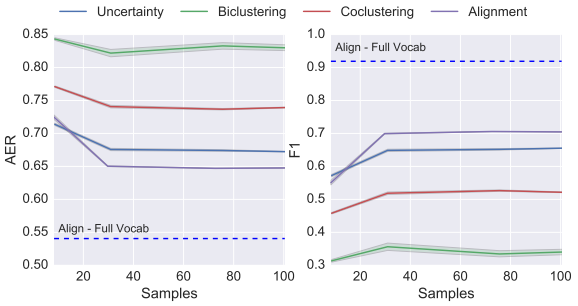


Figure 2: Arpabet test results as a function of number of perturbations used. Shown are mean plus confidence bounds over 5 repetitions. **Left**: Alignment Error Rate, **Right**: F1 over edge prediction.

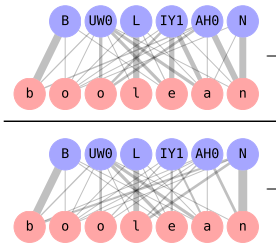
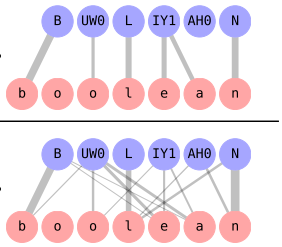
Raw Dependencies	Explanation Graph
	

Table 1: Inferred dependency graphs before (left) and after (right) explanation selection for the prediction: *boolean* $\mapsto$ `B UW0 L IY1 AH0 N`, in independent runs with large (top) and small (bottom) clustering parameter k.

for our method. The setting is as follows: given an input-output pair from the `cmudict` "black-box", we use our method to infer dependencies between characters in the input and phonemes in the output. Since locality in this context is morphological instead of semantic, we produce perturbations selecting n words randomly from the intersection of the `cmudict` vocabulary and the set of words with edit distance at most 2 from the original word.

To evaluate the inferred dependencies, we randomly selected 100 key-value pairs from the dictionary and manually labeled them with character-to-phoneme alignments. Even though our framework is not geared to produce pairwise alignments, it should nevertheless be able to recover them to a certain extent. To provide a point of reference, we compare against a (strong) baseline that is tailored to such a task: a state-of-the-art unsupervised word alignment method based on Monte Carlo inference (Tiedemann and Östling, 2016). The results in Figure 2 show that the version of our method that uses the uncertainty clustering performs remarkably close to the alignment system, with an alignment error rate only ten points above an oracle version of this system that was trained on the *full* arpabet dictionary (dashed line). The raw and partitioned explanations provided by our method for an example input-output pair are shown in Table 1, where the edge widths correspond to the estimated strength of dependency. Throughout this work we display the nodes in the same lexical order of the inputs/outputs to facilitate reading, even if that makes the explanation chunks less visibly discernible. Instead, we sometimes provide an additional (sorted) heatplot of dependency values to show these partitions.

5.3 Machine Translation

In our second set of experiments we evaluate our explanation model in a relevant and popular sequence-to-sequence task: machine translation. As black-boxes, we use three different methods for translating English into German: (i) Azure's Machine Translation system, (ii) a Neural MT model, and (iii) a human (native speaker of German). We provide details on all three systems in the supplement. We translate the same English sentences with all three methods, and explain their predictions using SOCRAT. To be able to generate sentences with similar language and structure as those used to train the two automatic systems, we use the monolingual English side of the WMT14 dataset to train the variational autoencoder described in Section 4.1. For every explanation instance, we sample $S = 100$ perturbations and use the black-boxes to translate them. In all cases, we use the same default SOCRAT configurations, including the robust partitioning method.

In Figure 3, we show the explanations provided by our method for the predictions of each of the three systems on the input sentence *"Students said they looked forward to his class"*. Although the three black-boxes all provided different translations, the explanations show a mostly consistent clustering around the two phrases in the sentence, and in all three cases the cluster with the highest cut value (i.e. the most relevant explanative chunk) is the one containing the subject. Interestingly, the

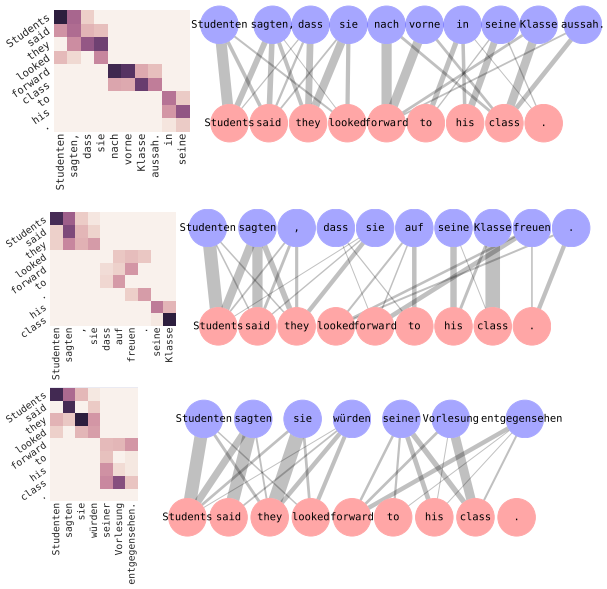


Figure 3: Explanations for the predictions of three Black-Box translators: Azure (top), NMT (middle) and human (bottom). Note that the rows and columns of the heatmaps are permuted to show explanation *chunks* (clusters).

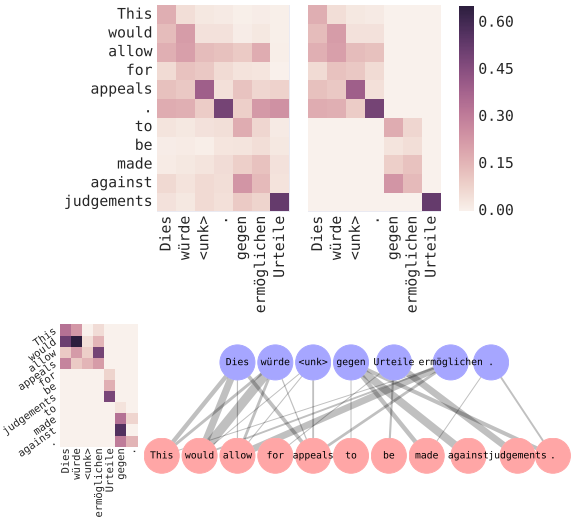


Figure 4: **Top**: Original and clustered attention matrix of the NMT system for a given translation. **Bottom**: Dependency estimates and explanation graph generated by SOCRAT with with $S = 100$.

dependency coefficients are overall higher for the human than for the other systems, suggesting more coherence in the translations (potentially because the human translated sentences in context, while the two automatic systems carry over no information from one example to the next).

The NMT system, as opposed to the other two, is not truly a black-box. We can *open the box* to get a glimpse on the true dependencies on the inputs used by the system at prediction time (the attention weights) and compare them to the explanation graph. The attention matrix, however, is dense and not normalized over target tokens, so it is not directly comparable to our dependency scores. Nevertheless, we can partition it with the coclustering method described in Section 4.3 to enforce group structure and make it easier to compare. Figure 4 shows the attention matrix and the explanation for an example sentence of the test set. Their overall cluster structure agrees, though our method shows conservatism with respect to the dependencies of the function words (*to*, *for*). Interestingly, our method is able to figure out that the <unk> token was likely produced by the word "appeals", as shown by the explanation graph.

It must be emphasized that although we display attention scores in various experiments in this work, we do so only for qualitative evaluation purposes. Our model-agnostic framework can be used on top of models that do not use attention mechanisms or for which this information is hard to extract. Even in cases where it is available, the explanation provided by SOCRAT might be complementary or even preferable to attention scores because: (a) being normalized on both directions (as opposed to only over source tokens) and partitioned, it is often more interpretable than a dense attention matrix, and (b) it can be retrieved *chunk-by-chunk* in decreasing order of relevance, which is especially important when explaining large inputs and/or outputs.

5.4 A (mediocre) dialogue system

So far we have used our method to explain (mostly) correct predictions of meaningful models. But we can use it to gain insights into the workings of flawed black-box systems too. To test this, we train a simple dialogue system on the OpenSubtitle corpus (Tiedemann, 2009), consisting of ∼14M two-step movie dialogues. As before, we use a sequence-to-sequence model with attention, but now we constrain the quality of the model, using only two layers, hidden state dimension of 1000 and no hyper-parameter tuning.

Input	Prediction
What do you mean it doesn't matter?	*I don't know*
Perhaps have we met before?	*I don't think so*
Can I get you two a cocktail?	*No, thanks.*

Table 2: "Good" dialogue system predictions.

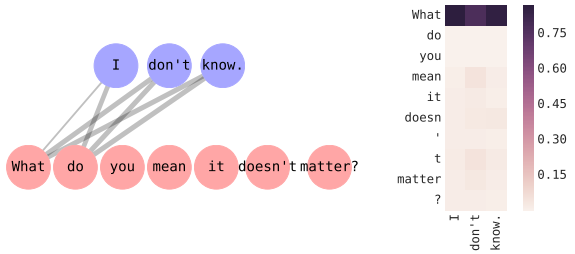


Figure 5: Explanation with $S = 50$ (left) and attention (right) for the first prediction in Table 2.

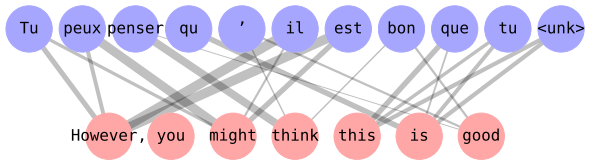


Figure 6: Explanation with $S = 50$ for the prediction of the biased translator.

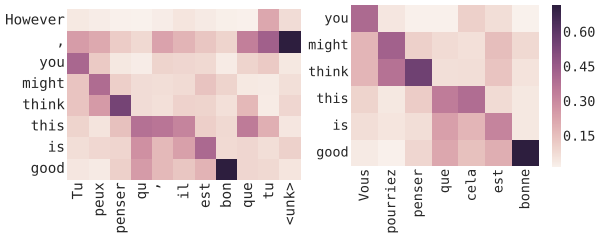


Figure 7: Attention scores on similar sentences by the biased translator.

Although most of the predictions of this model are short and repetitive (*Yes*/*No*/<unk> answers), some of them are seemingly meaningful, and might—if observed in isolation—lead one to believe the system is much better than it actually is. For example, the predictions in Table 2 suggest a complex use of the input to generate the output. To better understand this model, we rationalize its predictions using SOCRAT. The explanation graph for one such "good" prediction, shown in Figure 5, suggests that there is little influence of anything except the tokens *What* and *you* on the output. Thus, our method suggests that this model is using only partial information of the input and has probably memorized the connection between question words and responses. This is confirmed upon inspecting the model's attention scores for this prediction (same figure, right pane).

5.5 Bias detection in parallel corpora

Natural language processing methods that derive semantics from large corpora have been shown to incorporate biases present in the data, such as archaic stereotypes of male/female occupations (Caliskan et al., 2017) and sexist adjective associations (Bolukbasi et al., 2016). Thus, there is interest in methods that can detect and address those biases. For our last set of experiments, we use our approach to diagnose and explain biased translations of MT systems, first on a simplistic but *verifiable* synthetic setting, where we inject a pre-specified spurious association into an otherwise normal parallel training corpus, and then on an industrial-quality black-box system.

We simulate a biased corpus as follows. Starting from the WMT14 English-French dataset, we identify French sentences written in the informal register (e.g. containing the singular second person *tu*) and prepend their English translation with the word *However*. We obtain about 6K examples this way, after which we add an additional 1M examples that do not contain the word *however* on the English side. The purpose of this is to attempt to induce a (false) association between this adverb and the informal register in French. We then train a sequence-to-sequence model on this polluted data, and we use it to translate adversarially-chosen sentences containing the contaminating token. For example, given the input sentence "*However, you might think this is good*", the method predicts the translation "*Tu peux penser qu ' il est bon que tu <unk>*", which, albeit far from perfect, seems reasonable. However, using SOCRAT to explain this prediction (cf. Figure 6) raises a red flag: there is an inexplicable strong dependency between the function word *however* and tokens in the output associated with the informal register (*tu*, *peux*), and a lack of dependency between the second *tu* and the source-side pronoun *you*. The model's attention for this prediction (shown in Figure 7, left) confirms that it has picked up this spurious association. Indeed, translating the English sentence now without the prepended adverb

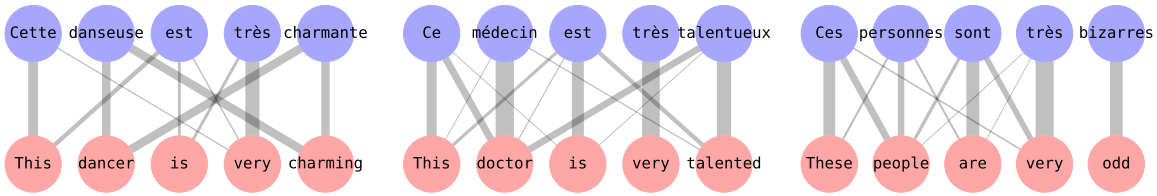


Figure 8: Explanations for biased translations of similar gender-neutral English sentences into French generated with Azure's MT service. The first two require gender declination in the target (French) language, while the third one, in plural, does not. The dependencies in the first two shed light on the cause of the biased selection of gender in the output sentence.

results in a switch to the formal register, as shown in the second plot in Figure 7.

Although somewhat contrived, this synthetic setting works as a litmus test to show that our method is able to detect *known* artificial biases from a model's predictions. We now move to a real setting, where we investigate biases in the predictions of an industrial-quality translation system. We use Azure's MT service to translate into French various simple sentences that lack gender specification in English, but which require gender-declined words in the output. We choose sentences containing occupations and adjectives previously shown to exhibit gender biases in linguistic corpora (Bolukbasi et al., 2016). After observing the choice of gender in the translation, we use SOCRAT to explain the output.

In line with previous results, we observe that this translation model exhibits a concerning preference for the masculine grammatical gender in sentences containing occupations such as *doctor, professor* or adjectives such as *smart, talented*, while choosing the feminine gender for *charming, compassionate* subjects who are *dancers* or *nurses*. The explanation graphs for two such examples, shown in Figure 8 (left and center), suggest strong associations between the gender-neutral but stereotype-prone source tokens (*nurse, doctor, charming*) and the gender-carrying target tokens (i.e. the feminine-declined *cette, danseuse, charmante* in the first sentence and the masculine *ce, médecin, talenteux* in the second). While it is not unusual to observe interactions between multiple source and target tokens, the strength of dependence in some of these pairs (*charming→danseuse*, *doctor→ce*) is unexplained from a grammatical point of view. For comparison, the third example—a sentence in the plural form that does not involve choice of grammatical gender in French—shows comparatively much weaker associations across words in different parts of the sentence.

6 Discussion

Our model-agnostic framework for prediction interpretability with structured data can produce reasonable, coherent, and often insightful explanations. The results on the machine translation task demonstrate how such a method yields a partial view into the inner workings of a black-box system. Lastly, the results of the last two experiments also suggest potential for improving existing systems, by questioning seemingly correct predictions and explaining those that are not.

The method admits several possible modifications. Although we focused on sequence-to-sequence tasks, SOCRAT generalizes to other settings where inputs and outputs can be expressed as sets of features. An interesting application would be to infer dependencies between textual and image features in image-to-text prediction (e.g. image captioning). Also, we used a VAE-based sampling for object perturbations but other approaches are possible depending on the nature of the domain or data.

Acknowledgments

We thank the anonymous reviewers for their helpful suggestions regarding presentation and additional experiments, and Dr. Chantal Melis for valuable feedback. DAM gratefully acknowledges support from a CONACYT fellowship and the MIT-QCRI collaboration.

References

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):1–46.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation By Jointly Learning To Align and Translate. *Iclr 2015*, pages 1–15.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *NIPS*, (Nips):4349—-4357.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. *Iclr*, pages 1–13.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science (80-.).*, 356(6334):183–186.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible Models for HealthCare : Predicting Pneumonia Risk and Hospital 30-day Readmission. *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '15*, pages 1721–1730.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2015. Listen, attend and spell. *arXiv Prepr.*, pages 1–16.

Inderjit s. Dhillon. 2001. Co-clustering documents and words using Bipartite spectral graph partitioning. *Proc 7th ACM SIGKDD Conf*, pages 269–274.

Finale Doshi-Velez and Been Kim. 2017. A Roadmap for a Rigorous Science of Interpretability. *ArXiv e-prints*, (Ml):1–12.

Neng Fan, Qipeng P. Zheng, and Panos M. Pardalos. 2012. Robust optimization of graph partitioning involving interval uncertainty. In *Theor. Comput. Sci.*, volume 447, pages 53–61.

M. R. Garey, D. S. Johnson, and L. Stockmeyer. 1976. Some simplified NP-complete graph problems. *Theor. Comput. Sci.*, 1(3):237–267.

Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. *Iclr*, (Ml):1–14.

G. Klein, Y. Kim, Y. Deng, J. Senellert, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

Yuval Kluger, Ronen Basri, Joseph T. Chang, and Mark Gerstein. 2003. Spectral biclustering of microarray data: Coclustering genes and conditions.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing Neural Predictions. In *EMNLP 2016, Proc. 2016 Conf. Empir. Methods Nat. Lang. Process.*, pages 107–117.

Zachary C Lipton. 2016. The Mythos of Model Interpretability. *ICML Work. Hum. Interpret. Mach. Learn.*, (Whi).

Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*.

D J Rezende, S Mohamed, and D Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. *Proc. 31st ...*, 32:1278–1286.

Danilo Jimenez Rezende and Shakir Mohamed. 2015. Variational Inference with Normalizing Flows. *Proc. 32nd Int. Conf. Mach. Learn.*, 37:1530–1538.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proc. 22Nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, KDD '16, pages 1135–1144, New York, NY, USA. ACM.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. *Proc. Conf. Empir. Methods Nat. Lang. Process.*, (September):379–389.

Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. 2016. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. (Nips):1–5.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. 2016. Ladder Variational Autoencoders. *NIPS*, (Nips).

Jörg Tiedemann. 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In N. Nicolov, G. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Adv. Nat. Lang. Process.*, pages 237—-248. John Benjamins, Amsterdam/Philadelphia.

Jörg Tiedemann and Robert Östling. 2016. Efficient Word Alignment with Markov Chain Monte Carlo. *Prague Bull. Math. Linguist.*, (106):125–146.