

Learning to Select Prototypical Parts for Interpretable Sequential Data Modeling

Yifei Zhang, Neng Gao*, Cunqing Ma

Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{zhangyifei, gaoneng, macunqing}@iie.ac.cn

Abstract

Prototype-based interpretability methods provide intuitive explanations of model prediction by comparing samples to a reference set of memorized exemplars or typical representatives in terms of similarity. In the field of sequential data modeling, similarity calculations of prototypes are usually based on encoded representation vectors. However, due to highly recursive functions, there is usually a non-negligible disparity between the prototype-based explanations and the original input. In this work, we propose a Self-Explaining Selective Model (SESM) that uses a linear combination of prototypical concepts to explain its own predictions. The model employs the idea of case-based reasoning by selecting sub-sequences of the input that mostly activate different concepts as prototypical parts, which users can compare to sub-sequences selected from different example inputs to understand model decisions. For better interpretability, we design multiple constraints including diversity, stability, and locality as training objectives. Extensive experiments in different domains demonstrate that our method exhibits promising interpretability and competitive accuracy.

Introduction

Deep neural networks have been widely employed for analyzing sequential data in real world, such as electrocardiogram (ECG), event streams, and natural language text. State-of-the-art methods generally leverage CNN, RNN, or other hybrid networks for sequence modeling. To meet the growing demand on explaining model predictions from black-box deep neural networks, more and more researchers resort to interpretable machine learning (IML) methods.

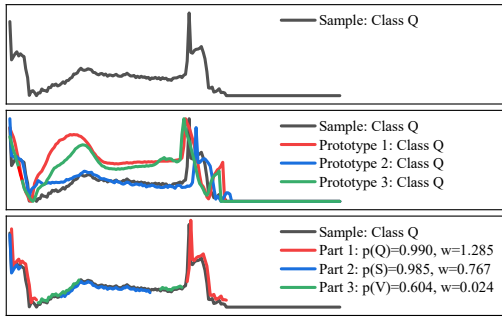
IML methods for sequential data can be roughly divided into three categories, including *post-hoc methods*, *attention-based methods*, and *prototype-based methods* (Samek et al. 2021). Post-hoc methods (Jacovi, Shalom, and Goldberg 2018) are the only options to provide *a posteriori* explanations for trained models by assessing the impact of different input features and approximating the decision-making process. Attention-based methods and prototype-based methods explicitly reveal some significant information for model prediction in different ways, resulting in model-intrinsic interpretability. Specifically, attention-based methods (Wang

et al. 2019) introduce additional parameters and specific model structure to weigh the importance of sequence elements, namely attention mechanism, for giving insight into models. Prototype-based methods (Li et al. 2018), derived from case-based reasoning (Kim, Rudin, and Shah 2014), design machine learning models that select or create a set of representative instances as prototypes (also named as concepts). The model tries to find several prototypes that closely resemble an input for prediction. By inspecting the selected prototypes, the explanations are more intuitively understandable for laypersons.

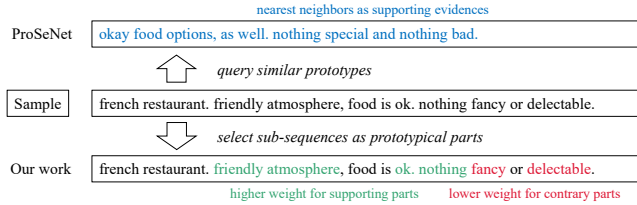
However, recent efforts are often faced with the following challenges. Post-hoc explanations are shown to be incorrect or incomplete, since the approximation may not always reflect the real model structure (Laugel et al. 2019; Lakkaraju and Bastani 2020). The interpretability of attention mechanism remains controversial (Jain and Wallace 2019; Wiegrefe and Pinter 2019). Although attention mechanism has been widely applied and helps models attend to more significant elements or features, some have questioned that attention-based explanations are unreliable or unfaithful (Serrano and Smith 2019; Bai et al. 2021). Existing prototype-based methods mostly learn a fixed set of representative vectors as prototypes, and an input is represented by one or more prototypes as explanation in practice. For modeling sequences with rich information, state-of-the-art methods (Ming et al. 2019; Chen et al. 2019; Arik and Pfister 2020) need to maintain a rather large number of prototypes to achieve reliable performance (Hong, Baek, and Wang 2020). Moreover, the similarity-based measurements for finding prototypes are applied on hidden representations generated by encoders with highly recursive functions, thus the direct relation between prototypes and the input sequence might still be hard to understand for human. Overall analysis of the provided prototypes as well as the input sequence is required, which deviates from the interpretability assumption to some extent.

In this work, we suggest enhancing the prototype-based interpretability for sequential data with self-selective prototypical parts, and design a Self-Explaining Selective Model (SESM). Rather than maintaining a relatively large amount of instances and find the nearest neighbor for explanation, SESM explains by selecting sub-sequences that represent disentangled concepts of the input sequence as pro-

*Corresponding author.



(a) ECG signal classification. Top: signal sample. Center: prototypes of ProSeNet. Bottom: prototypical parts of proposed SESM.



(b) Review semantic classification. Color distinguishes words from different sub-sequences.

Figure 1: Difference between similarity-based prototypes and the proposed self-selective prototypical parts.

prototypical parts. Motivated by the general framework of the self-explaining neural network (SENN) (Alvarez-Melis and Jaakkola 2018), SESM leverages a modified multi-head self-attention mechanism as conceptizer to select prototypical parts, where each head tends to select the sub-sequence of the input that mostly activates a specific concept. Then, the prototypical parts of an input are encoded separately and aggregated linearly as the final modeling result. Note that since the concepts are represented in the form of sub-sequences, they can be encoded by any existing sequence modeling methods and explained via prototyping (i.e., create a small set of sub-sequences from training set with similar concepts). As instantiated in Figure 1, the prototypical parts manifest as supporting or contrary evidences for model prediction according to their assigned weights, where the supporting evidence shows the decisive sub-sequences of the input, while the contrary evidence represents the part of the input that could lead the model to output a different class. For better interpretability, we design several learning criteria to impose unsupervised disentanglement (Rudin et al. 2021) in the end-to-end training process, including diversity, stability, and locality, where diversity reduces redundancy of concepts, stability provides conceptual unity, and locality prevents prototype parts from degenerating into prototypes. Hence, the model is inherently interpretable that can provide straightforward and brief explanations for model predictions. Moreover, the selection of prototypical parts does not require storage of raw data from the training set, which could be a stepping stone towards GDPR compliance. Our contributions of proposing SESM can be summarized as follows:

- SESM is an inherently interpretable sequence model, which selects sub-sequences representing disentangled concepts of an input as prototypical parts for explaining its own predicting process.
- SESM is end-to-end trainable with our designed learning criteria for the training process, which impose constraints on extracting straightforward and brief explanations, without memorizing a rather large amount of raw data from the training set.
- SESM shows comparable effectiveness and outperforms baseline methods in terms of interpretability based on our extensive experiments on various domains.

Related Work

Compared with post-hoc and attention-based explanations, prototype-based explanations are more easily understandable for human through case-based reasoning (Kim, Rudin, and Shah 2014; Alvarez-Melis and Jaakkola 2018). PrototypeDL (Li et al. 2018) maintains multiple representative vectors as prototypes, and the similarities between an input and the prototypes are concatenated as a representation vector for classification. The prototype vectors are required to be as close as possible to a real training instance in latent space, so that humans may inspect the most similar training instances to the input in the latent space for explanation. SENN (Alvarez-Melis and Jaakkola 2018) automatically extracts representative concepts from inputs as prototypes, and samples instances that maximally activate each concept for explanation. ProSeNet (Ming et al. 2019) provides prototype-based explanations for sequential data with the similar idea as PrototypeDL. Instead of prototyping the entire sequences, SelfExplain (Rajagopal et al. 2021) and SCNpro (Ni et al. 2021) prototype local segments of sequences for more accurate and understandable explanation.

Our work differs from existing work in the following aspects. First, the framework of PrototypeDL and ProSeNet requires encoding the entire sequences for similarity-based prototype query. The sequence-level encoding of input inevitably involves irrelevant or opposite elements, while SESM learns to select sub-sequences with disentangled concepts, which can provide more fine-grained explanations. Second, the way SENN extracts concepts from the input is not interpretable, since the conceptizers are black-box neural networks. Instead, SESM generates concepts with selective prototypical parts, which is a part of the original sequence to be immediately understandable. SelfExplain also successfully addresses this issue by leveraging constituency-based parse trees to generate text spans of words and phrases from the input sentence as concepts, while at a cost of limited application scenario of neural language classification. Third, the sub-sequences for prototyping claimed by SCNpro is actually local segments of continuous elements, while SESM works on actual sub-sequences that can be composed of discontinuous elements, which can help capture distant elements representing similar concepts with less amount of prototypes. Moreover, we would like to stress that the selective prototypical parts are created solely based on the selective actions, in order to completely eliminate the effect

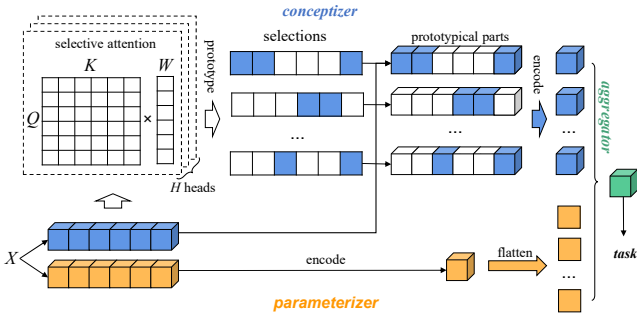


Figure 2: The overall architecture of our proposed SESM. Squares denote scalars and cubes denote vectors.

of discarded elements, and can be encoded as concepts with any existing sequence modeling methods.

Methodology

Model Architecture

For inherently interpretability, SESM is comprised of three main modules following the self-explaining framework of SENN, including *conceptizer*, *parameterizer* and *aggregator*. The conceptizer selects multiple sub-sequences from the raw input as prototypical parts representing disentangled concepts, and the parameterizer determines which concepts are of greater importance for model prediction by assigning different weights. Then, the aggregator linearly combines the prototypical parts to obtain the final representation of the input sequence. The detailed architecture of SESM is shown in Figure 2. Let $X = \{x_1, x_2, \dots, x_N\}$ denote an input sequence with N elements. Next, we will formalize the detailed implementation of each module respectively.

The **conceptizer** \mathcal{C} aims to create H sub-sequences of X with disentangled concepts $\{X_1, X_2, \dots, X_H\}$. Specifically, a vector of selective actions $s_h = \mathcal{C}_h(X) = \{0, 1\}^N$ is generated for each prototypical part X_h based on the selective mechanism, indicating which elements X_h is composed of. By combining H different selection vectors, the output of conceptizer can be denoted as a selection matrix $\mathcal{C}(X) = \{s_h\}_{h=1}^H = \{0, 1\}^{H \times N}$, where $s_{h,i} = 1$ indicates that the h -th prototypical part includes the i -th element of X .

Geng et al. (2020) have proposed the selective self-attention network (SSAN), which is an ideal base method for realizing the function of our conceptizer with selective mechanism. Given input sequence X , the SSAN embeds and projects X into three d_h -dimensional matrices queries, keys and values $Q, K, V \in \mathbb{R}^{N \times d}$. Then, SSAN applies the commonly used dot-product attention (Vaswani et al. 2017) to obtain the selective attention weights based on element pairs:

$$\text{ATT}(Q, K) = \text{Gumbel-Sigmoid} \left(QK^T / \sqrt{d_h} \right), \quad (1)$$

where the Gumbel-Sigmoid operation is a reparametrization trick for training non-differentiable model with selective operations, and the pair-wise attention is applied on V

as $\text{ATT}(Q, K) \times V$. By stacking multiple Q, K, V combinations, namely multi-head attention, the final modeling result of X is able to include different perspectives of information in parallel.

However, the operation of applying self-attention mechanism leads to relatively weak model interpretability. The dot-product attention is based on entangled pair-wise information of elements in Q and K , and is applied on V through matrix multiplication, thus the attentive information cannot be directly associated to a small set of raw elements from the input sequence for intuitive explanation.

For better interpretability, we introduce an additional matrix for projection $W \in \mathbb{R}^{N \times 1}$ into Eq. 1. By grouping the pair-wise attentions for each element in rows, W squeezes the pair-wise attentions into element-wise attentions before binarized:

$$C_h(X) = \text{Gumbel-Sigmoid} \left(QK^T W / \sqrt{d_h} \right). \quad (2)$$

We can then extract sub-sequences as human-friendly prototypical parts for explanation with the element-wise selective attention. After that, the concepts of an input is encoded according to the selected prototypical parts. Let Enc denote an arbitrary encoder, the encoded concept c_h of the h -th prototypical part can be denoted as $c_h = Enc(X_h) = Enc(X \otimes s_h)$, where \otimes denotes applying selection s_h on input X .

The **parameterizer** \mathcal{P} aims to decide the contributions of different concepts for model prediction. According to SENN, the entries of parameters for interpretability should be non-negative. The parameterizer models the entire sequence with stacked CNNs followed by an MLP for projection to weigh each prototypical part X_h , where the output of $\mathcal{P}(X) = \{p_h\}_{h=1}^H$ is an H -dimensional vector activated by Softplus.

The **aggregator** \mathcal{G} aggregates the concepts of prototypical parts c_h and their corresponding non-negative weights p_h for an overall representation vector for downstream tasks. The aggregation process is additive on the encoded concepts for interpretability. As introduced above, a prototypical part X_h is represented by a binary vector that indicates the selective action of each entry. Thus, the aggregating result, i.e., the overall modeling result of SESM, can be formalized as:

$$\text{SESM}(X) = \mathcal{G}(\mathcal{P}(X)\mathcal{C}(X)) = \sum_{h=1}^H p_h c_h. \quad (3)$$

Learning Objective

The learning objectives of SESM should meet the demand of both utility and interpretability. Let $f_{\text{task}}(\cdot)$ denote a network for a downstream task (e.g., ECG signal classification, sentence semantic classification), and y denote the corresponding ground-truth label in the training set. The loss of utility is then $\mathcal{L}_{\text{task}}(f_{\text{task}}(\text{SESM}(X)), y)$, e.g., negative log likelihood loss and cross-entropy loss. For interpretability, we design three learning criteria as regularization for the conceptizer and parameterizer. Next, we describe the proposed criteria along with the reasons to adopt them.

Diversity The prototypical parts should comprise different elements of a sequence as well as representing different perspectives of information for disentanglement, in order to reduce redundancy (multiple prototypical parts represent similar concepts) and incompleteness (a single prototypical part does not cover all necessary elements) of explanation. Accordingly, we design the diversity regularization to constrain that different prototypical parts select different elements in the input X . Inspired by ProSeNet, we leverage L2 distance to design the loss function of diversity with threshold $d_{\min} = 2$:

$$\mathcal{L}_d = \sum_{i=1}^{H-1} \sum_{j=i+1}^H \left[\text{RELU} (d_{\min} - \|s_i - s_j\|^2) \right]. \quad (4)$$

Stability For better disentanglement and stability, we require each head of the conceptizer to focus on a single concept. Specifically, the encoded representations c_h of prototypical parts selected by the same head of conceptizer \mathcal{C} are similar. The regularization is implemented by minimizing the pair-wise cosine distance of encoded concepts from the same head at batch level. Formally, we have:

$$\mathcal{L}_s = \sum_{h=1}^H \sum_{i=1}^{B-1} \sum_{j=i+1}^B \left[1 - \cos(c_i^h, c_j^h) \right], \quad (5)$$

where B denotes the batch size.

Locality During experiment, we noticed that the conceptizer would occasionally fail, when some of the heads select all elements of the original sequence X and the others select none. To tackle this problem and further encourage diversity in the prototypical parts, we introduce locality loss to penalize heads that select an excessive amount of components:

$$\mathcal{L}_l = \sum_{h=1}^H \frac{1}{N} \sum_{i=1}^N s_{h,i}, \quad (6)$$

To sum up, the overall learning objective is:

$$\mathcal{L} = \mathcal{L}_{task} + \lambda_d \mathcal{L}_d + \lambda_s \mathcal{L}_s + \lambda_l \mathcal{L}_l, \quad (7)$$

where λ weighs each regularization term.

Model Interpretability

The interpretability of our proposed SESM is in two fold. First, SESM selects several prototypical parts representing disentangled concepts and assign different weights to them, which illustrates what concepts a sequence is comprised of and the corresponding importance for model prediction, resulting in model-intrinsic interpretability via unsupervised disentanglement. Then, the selected prototypical parts can be further interpreted via prototyping. Prototype-based explanations can help users to understand model prediction through case-based reasoning. By sampling the prototypical parts of instances in the training set that maximally activate a concept, users can intuitively inspect the concept by comparing the selected prototypical part from the input with the most influential prototypes, which provides fine-grained prototype-based interpretability.

Experiments

Experimental Settings

We compare SESM with multiple open source baselines, including black-box methods without interpretability LSTM and CNN, attention-based method named SAN (Lin et al. 2017), and methods with prototype-based interpretability ProSeNet and SelfExplain. Since the original implementation of ProSeNet is not open-source, we migrate the reproduced TensorFlow version on github¹ for comparison. SelfExplain relies on constituency-based parse trees generated by pre-trained neural language models, thus it is only applied for comparison on natural language tasks. For fairness, we do not compare SESM with the most recent work SCN-pro, since it is not open source by far. The codes for implementing SESM and our reproduction of ProSeNet will be released on Github². The experiments are conducted with PyTorch 1.8 on two NVIDIA Tesla V100 16GB GPUs. The AdamW optimizer (Loshchilov and Hutter 2019) is employed for all experiments, which improves the generalization performance of the commonly used Adam optimizer. The experiment results are averaged on 10 runs with different random seeds.

We evaluate the effectiveness of our proposed SESM on tasks from various domains. The accuracy and macro-averaged precision and recall are employed for measuring accuracy. To the best of our knowledge, there is no standardized method for quantitatively assessing interpretability, thus we follow existing work (Li et al. 2018; Ming et al. 2019; Ni et al. 2021) to highlight qualitative case studies, and migrate the area over perturbation curve (AOPC) (Nguyen 2018; Chen, Zheng, and Ji 2020) for quantitative analysis. AOPC is utilized as a counterfactual assessment of word-level explanations for text classification, by measuring the average change in the prediction probability on the predicted class after deleting top-scored words. For prototype-based methods, we migrate AOPC to quantify the prototype-based interpretability of models by measuring the probability drop of the predicted class after deleting the most relevant prototypes (or prototypical parts in SESM):

$$\text{AOPC} = \frac{1}{H-1} \left\langle \sum_{h=1}^{H-1} f(\mathbf{x}) - f(\mathbf{x}_{\setminus 1..h}) \right\rangle, \quad (8)$$

where $f(\mathbf{x}_{\setminus 1..h})$ is the probability on the predicted class *without* the top h relevant prototypes, and $\langle \cdot \rangle$ denotes the average over samples. A larger AOPC indicates that the model shifts its prediction more drastically and implies that the most contributing prototypes for the final prediction are of greater significance and less redundancy.

Next, we will briefly introduce the leveraged tasks and case studies about sequence modeling respectively.

Task 1: ECG Signal Classification We first evaluate SESM on real-valued ECG signal classification task using the pre-processed MIT-BIH Arrhythmia dataset³ as exist-

¹<https://github.com/rgmyr/tf-ProSeNet>

²<https://github.com/iezyf/SESM>

³<https://www.kaggle.com/shayanfazeli/heartbeat>

Method	Acc.	Avg.P	Avg.R	AOPC
LSTM	0.985	0.923	0.851	-
CNN	0.984	0.931	0.900	-
SAN	0.940	0.734	0.722	0.176
ProSeNet	0.978	0.938	0.844	0.186
SESM	0.982	0.931	<u>0.893</u>	0.232
SESM +w	0.987	<u>0.932</u>	0.921	0.230

Table 1: Performance on MIT-BIH dataset. “+w” denote training with class weights.

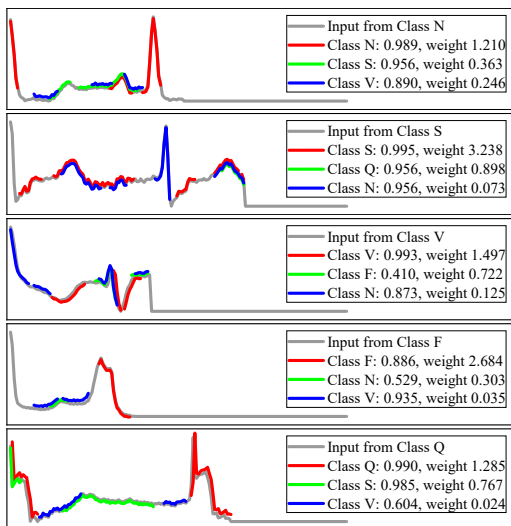
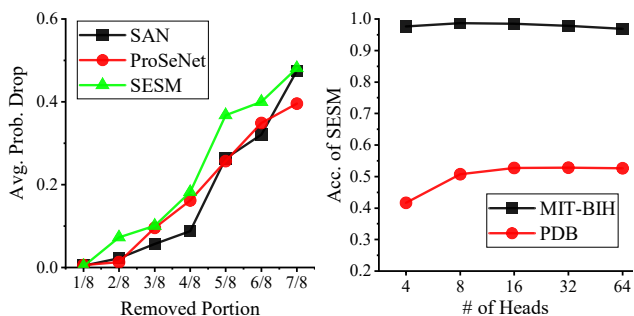


Figure 3: Case study on MIT-BIH dataset. Prototypical parts are marked with different colors on the input sequence. The legend shows the prediction probability based solely on the prototypical part, as well as the corresponding weight for aggregation.

ing work (Ming et al. 2019; Ni et al. 2021). The MIT-BIH dataset consists of annotated ECG signals of heartbeats from 5 significantly skewed classes, including Normal (N), Atrial Premature (S), Premature ventricular contraction (V), Fusion of ventricular and normal (F), Fusion of paced and normal (Q), where the amount of instances from each class is [90589, 2779, 7236, 803, 8039](Kachuee, Fazeli, and Sarrafzadeh 2018). In this experiment, we first employ a Conv1d layer with kernel size 10 as an embedding operation to project the time serial scalars in sequences into hidden representations, and use the segments as the smallest units for modeling and explanation. Since the reproduced version of ProSeNet often fails when training with class weights, we compare all baseline methods without class weights.

Task 2: Protein Family Classification The second task is protein family classification using PDB dataset⁴ provided by Protein Data Bank. The PDB dataset contains protein sequences in various lengths composed of 20 standard amino acids, and can be grouped into families. We filtered out se-

⁴<https://www.kaggle.com/shahir/protein-data-set>



(a) Counterfactual assessment. (b) Impact of heads H .

Figure 4: Detailed analysis on MIT-BIH dataset.

Method	Acc.	Avg.P	Avg.R	AOPC
LSTM	0.526	0.381	0.333	-
CNN	0.538	0.399	0.322	-
SAN	0.517	0.380	0.311	0.130
ProSeNet	0.489	0.334	0.277	0.092
SESM	0.526	0.389	0.333	<u>0.135</u>
SESM +w	<u>0.525</u>	<u>0.381</u>	<u>0.317</u>	0.153

Table 2: Performance on PDB dataset.

quences whose length are less than 50 and clipped the sequences from the top 100 largest families with a maximal length of 512, following the settings of ProSeNet.

Task 3: Text Sentiment Classification The third task is text sentiment classification on the widely used dataset Yelp Reviews⁵, including binary classification (YelpP) and five-score classification (YelpF). We tokenize the reviews with `Torchtext` and only use those with less than 25 words in the experiments. The word embeddings are initialized with the 300D GloVe 6B pre-trained vectors (Pennington, Socher, and Manning 2014).

Task 4: Natural Language Inference We evaluate SESM on natural language inference task with the SNLI (Bowman et al. 2015) dataset, which aims to classify the semantic relation between a pair of sentences into three categories, including neutral (N), contradiction (C) and entailment (E). We follow the experimental settings of existing work SAN (Lin et al. 2017) and SSAN (Geng et al. 2020). The sentences of premise and hypothesis are encoded by the model, denoted as s_p and s_h respectively. Then, the classification is based on a multi-layer perceptron with two fully connected layers activated by ReLU on the concatenation of $[s_p, s_h, s_p - s_h, s_p \cdot s_h]$.

Since SelfExplain is originally designed as a classification method for single sentences, we tried to adapt SelfExplain to fit the pair-wise natural language inference task. Specifically, the original SelfExplain encodes a complete sentence into a representation vector s , and find the hidden representations of concepts (words or phrases) that are

⁵<https://www.yelp.com/dataset>

Method	#	Review Text	Label (w)
Raw Input		french restaurant. friendly atmosphere, food is ok. nothing fancy or delectable.	3
ProSeNet	1	okay food options, as well. nothing special and nothing bad.	3 (0.45)
	2	nothing special. food is okay, drive through is fast and friendly.	3 (0.31)
SelfExplain	1	french restaurant. friendly atmosphere, food is ok. nothing fancy or delectable.	3 (N/A)
	2	french restaurant. friendly atmosphere, food is ok . nothing fancy or delectable.	3 (N/A)
Our SESM	1	french restaurant. friendly atmosphere , food is ok . nothing fancy or delectable.	3 (0.83)
	2	french restaurant. friendly atmosphere, food is ok. nothing fancy or delectable .	5 (0.16)

Figure 5: Case study on Yelp dataset. Column # distinguishes different explanatory units. Color distinguishes words from different heads or sub-sequences.

Method	#	Premise	Hypothesis	Label (w)
Raw Input		a man in a green beret enjoying espresso with a friend.	two people are having drinks.	E
ProSeNet	1	two older gentlemen having coffee or tea and a serious discussion.	two men are drinking and having a talk.	E (0.34)
	2	a man standing in front of a building on the phone as two men to the side pain on the side.	a guy near a building stands by two other men.	E (0.21)
SelfExplain	1	a man in a green beret enjoying espresso with a friend.	two people are having drinks .	E (N/A)
	2	a man in a green beret enjoying espresso with a friend.	two people are having drinks.	E (N/A)
Our SESM	1	a man in a green beret enjoying espresso with a friend .	two people are having drinks .	E (0.76)
	2	a man in a green beret enjoying espresso with a friend.	two people are having drinks .	C (0.21)

Figure 6: Case study on SNLI dataset.

Metric	Method	YelpP	YelpF	SNLI
Acc.	LSTM	0.953	0.704	0.786
	CNN	0.957	0.704	0.810
	SAN	0.952	0.692	0.796
	ProSeNet	0.957	0.694	0.741
	SelfExplain	0.951	0.689	0.728
	SESM	0.949	0.695	0.803
AOPC	SAN	0.363	0.257	0.267
	ProSeNet	0.375	0.222	0.208
	SelfExplain	<u>0.379</u>	<u>0.268</u>	<u>0.287</u>
	SESM	0.480	0.336	0.412

Table 3: Experimental results on NLP datasets.

most similar to s for explanation. By going through all possible pairs of concepts (generated by the constituency-based parse trees) from two sentences, we replace s with $[s_p, s_h, s_p - s_h, s_p \cdot s_h]$, and the explanatory representations are then based on the most matchable pair-wise phrases.

Experimental Results and Case Studies

Tables 1, 2 and 3 show the experiment results on the accuracy and the quantified interpretability of models for comparison. The **bold** values indicate the best results of the interpretable models for each entry, and the underlined values indicate the second best results. SESM shows comparable accuracy on various tasks, while all models with interpretability are facing performance loss to various degrees. Nevertheless, SESM shows better interpretability on the counterfactual assessment AOPC, since it is designed to model disentangled concepts from different facets.

We illustrate the explanations on MIT-BIH dataset pro-

vided by SESM in Figure 3. For each class, we randomly sample an instance of ECG signal, and highlight three most important prototypical parts representing different concepts that would be classified into distinct classes (i.e., skipping the sub-sequences that would lead to the same result). Based on the selection of prototypical parts, SESM can directly mark the representative sub-sequences on the original input.

To better understand the explanations provided by different heads of SESM on MIT-BIH dataset, we illustrate the statistics of the predictive results in Figure 8 and instantiate 10 prototypical parts for each head in Figure 7. The prototypes are provided to three laypersons of ECG diagnosis for human evaluation of model interpretability. The average accuracy of human subjects increases from 0.543 to 0.691, which outperforms the situation of providing prototypes generated by ProSeNet (30 prototypes, increase from 0.543 to 0.610). Presented below is a succinct overview of the inference process adopted by the three non-experts, upon being presented with the prototypical parts and the corresponding input for each head.

The inference process starts with the most distinguishable heads. According to Figure 8, Heads 2, 3, 5 and 6 are with less ambiguity for the first four kinds of heartbeats N, S, F, and V, respectively. As shown in Figure 7, Head 2 focuses on the crests of ECG signals, as well as Head 8. The selected prototypical parts illustrate how the crests arranged in normal heartbeats. Specifically, some minor bumps between two major peaks are typical for a normal heartbeat. Head 3 generally attends to the pattern of abnormally elevation of waves just after the major peaks, which is clearly different from the N beats. Analogously, the prototypical parts selected by Head 4 also shown the elevation of waves but at a lower degree, so that the predictive results of Head 4 is

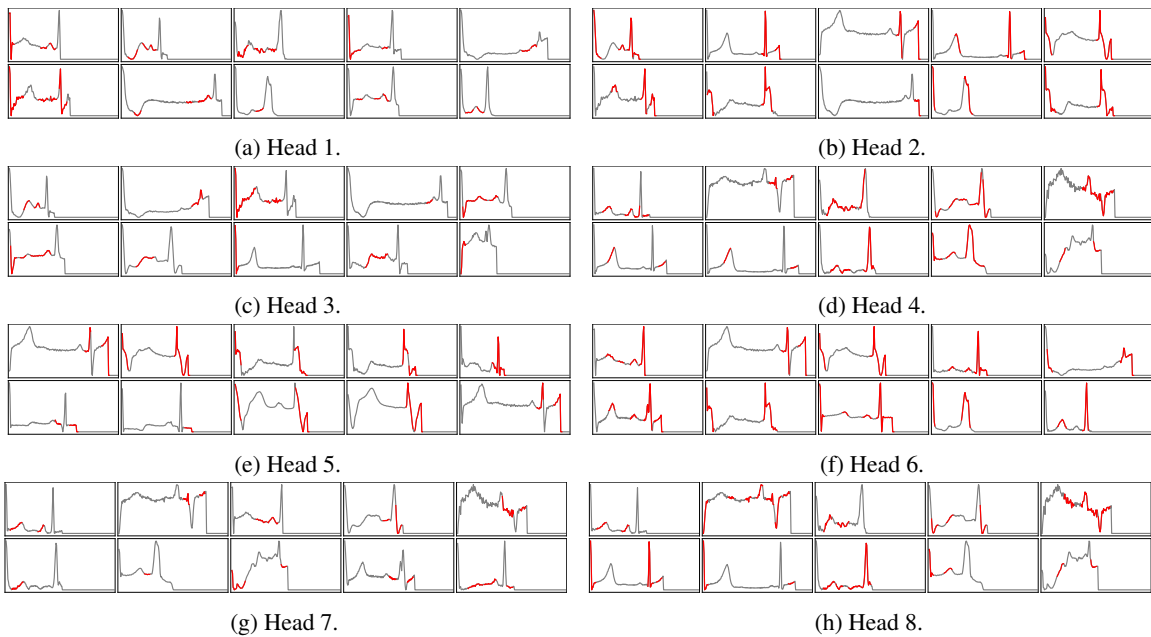


Figure 7: Prototyping for explaining prototypical parts selected by different heads on MIT-BIH dataset. Note that the inputs are randomly sampled from different classes, but the selected prototypical parts activates the same concept within each head.

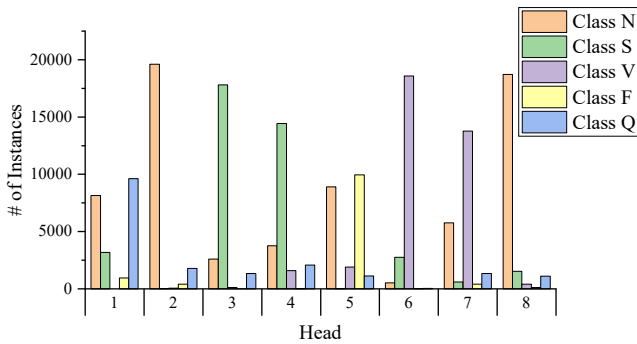


Figure 8: Statistics of the predictive results from different heads on MIT-BIH dataset.

less certain. Head 5 also focuses on the crests of ECG signals, but the selected prototypical parts are not as neat as those in Head 2, which are classified into Class F. Head 6 shows the pattern of a relatively slow decline of the peaks. Combined with Head 7, the evidences for forecasting Class V are clear to grasp. Finally, Head 1 simply attend to the pattern of a plateau followed by a cliff at the second peak as the evidence for Class Q.

Figures 5 and 6 show the case studies of NLP tasks on models with prototype-based interpretability. Compared with ProSeNet, SelfExplain and SESM provide concept-level prototypical parts as explanations, which is more intuitively understandable for laypersons. Compared with SelfExplain, SESM explicitly shows the weights of the prototypical parts for model prediction, instead of the post-hoc similarity-based analysis on the hidden representations. Be-

sides, the prototypical parts used in SESM are more comprehensive than SelfExplain, since SESM imposes constraints that the selected prototypical parts should be distinct with each other, which also helps to provide counterfactual explanations about how to flip the model prediction by given only a sub-sequence of the input.

We further analyze the performance of SESM with different number of concepts, i.e. the number of heads H . Figure 4b shows that SESM performs better with 8 heads for MIT-BIH dataset and with 16 heads for PDB dataset, whereas it does not significantly affect the performance to keep increasing the number of heads. According to our further analysis, some of the excessive heads would tend to be consistent with each other, thus the diversity loss would increase but has a minor impact on the performance.

Conclusion and Future Work

In this work, we propose a self-explaining selective model for interpretable sequence modeling named SESM, which selects sub-sequences from the raw input representing disentangled concepts as prototypical parts to provide more intuitively understandable explanations. It eliminates discrepancies between hidden layer representation vectors and original sequence of prototypes when understanding prototype-based interpretations. Experiments on various domains demonstrate that SESM consistently outperforms baseline interpretability methods with stronger impact on counterfactual assessment, while retaining comparable model accuracy with baseline methods with and without interpretability. Future work includes designing an aggregator that could leverage the higher-order interaction of features from different concepts, while retaining interpretability.

Acknowledgments

This work is supported by the National Key Research and Development Program of China.

References

- Alvarez-Melis, D.; and Jaakkola, T. S. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 7786–7795.
- Arik, S. Ö.; and Pfister, T. 2020. ProtoAttend: Attention-Based Prototypical Learning. *J. Mach. Learn. Res.*, 21: 210:1–210:35.
- Bai, B.; Liang, J.; Zhang, G.; Li, H.; Bai, K.; and Wang, F. 2021. Why Attentions May Not Be Interpretable? In Zhu, F.; Ooi, B. C.; and Miao, C., eds., *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, 25–34. ACM.
- Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In Márquez, L.; Callison-Burch, C.; Su, J.; Pighin, D.; and Marton, Y., eds., *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 632–642. The Association for Computational Linguistics.
- Chen, C.; Li, O.; Tao, D.; Barnett, A.; Rudin, C.; and Su, J. 2019. This Looks Like That: Deep Learning for Interpretable Image Recognition. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 8928–8939.
- Chen, H.; Zheng, G.; and Ji, Y. 2020. Generating Hierarchical Explanations on Text Classification via Feature Interaction Detection. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 5578–5593. Association for Computational Linguistics.
- Geng, X.; Wang, L.; Wang, X.; Qin, B.; Liu, T.; and Tu, Z. 2020. How Does Selective Mechanism Improve Self-Attention Networks? In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2986–2995. Association for Computational Linguistics.
- Hong, D.; Baek, S. S.; and Wang, T. 2020. Interpretable Sequence Classification Via Prototype Trajectory. *CoRR*, abs/2007.01777.
- Jacovi, A.; Shalom, O. S.; and Goldberg, Y. 2018. Understanding Convolutional Neural Networks for Text Classification. In Linzen, T.; Chrupala, G.; and Alishahi, A., eds., *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, 56–65. Association for Computational Linguistics.
- Jain, S.; and Wallace, B. C. 2019. Attention is not Explanation. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 3543–3556. Association for Computational Linguistics.
- Kachuee, M.; Fazeli, S.; and Sarrafzadeh, M. 2018. ECG Heartbeat Classification: A Deep Transferable Representation. In *IEEE International Conference on Healthcare Informatics, ICHI 2018, New York City, NY, USA, June 4-7, 2018*, 443–444. IEEE Computer Society.
- Kim, B.; Rudin, C.; and Shah, J. A. 2014. The Bayesian Case Model: A Generative Approach for Case-Based Reasoning and Prototype Classification. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 1952–1960.
- Lakkaraju, H.; and Bastani, O. 2020. "How do I fool you?": Manipulating User Trust via Misleading Black Box Explanations. In Markham, A. N.; Powles, J.; Walsh, T.; and Washington, A. L., eds., *AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020*, 79–85. ACM.
- Laugel, T.; Lesot, M.; Marsala, C.; Renard, X.; and Detryniecki, M. 2019. The Dangers of Post-hoc Interpretability: Unjustified Counterfactual Explanations. In Kraus, S., ed., *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 2801–2807. ijcai.org.
- Li, O.; Liu, H.; Chen, C.; and Rudin, C. 2018. Deep Learning for Case-Based Reasoning Through Prototypes: A Neural Network That Explains Its Predictions. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 3530–3537. AAAI Press.
- Lin, Z.; Feng, M.; dos Santos, C. N.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A Structured Self-Attentive Sentence Embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

- Ming, Y.; Xu, P.; Qu, H.; and Ren, L. 2019. Interpretable and Steerable Sequence Learning via Prototypes. In Teredesai, A.; Kumar, V.; Li, Y.; Rosales, R.; Terzi, E.; and Karypis, G., eds., *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 903–913. ACM.
- Nguyen, D. 2018. Comparing Automatic and Human Evaluation of Local Explanations for Text Classification. In Walker, M. A.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, 1069–1078. Association for Computational Linguistics.
- Ni, J.; Chen, Z.; Cheng, W.; Zong, B.; Song, D.; Liu, Y.; Zhang, X.; and Chen, H. 2021. Interpreting Convolutional Sequence Model by Learning Local Prototypes with Adaptation Regularization. In Demartini, G.; Zuccon, G.; Culpepper, J. S.; Huang, Z.; and Tong, H., eds., *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, 1366–1375. ACM.
- Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar; A meeting of SIGDAT, a Special Interest Group of the ACL*, 1532–1543. ACL.
- Rajagopal, D.; Balachandran, V.; Hovy, E. H.; and Tsvetkov, Y. 2021. SELFEXPLAIN: A Self-Explaining Architecture for Neural Text Classifiers. In Moens, M.; Huang, X.; Specia, L.; and Yih, S. W., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, 836–850. Association for Computational Linguistics.
- Rudin, C.; Chen, C.; Chen, Z.; Huang, H.; Semanova, L.; and Zhong, C. 2021. Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges. *CoRR*, abs/2103.11251.
- Samek, W.; Montavon, G.; Lapuschkin, S.; Anders, C. J.; and Müller, K. 2021. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. *Proc. IEEE*, 109(3): 247–278.
- Serrano, S.; and Smith, N. A. 2019. Is Attention Interpretable? In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2931–2951. Association for Computational Linguistics.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.
- Wang, J.; Liu, Q.; Liu, Z.; and Wu, S. 2019. Towards Accurate and Interpretable Sequential Prediction: A CNN & Attention-Based Feature Extractor. In Zhu, W.; Tao, D.; Cheng, X.; Cui, P.; Rundensteiner, E. A.; Carmel, D.; He, Q.; and Yu, J. X., eds., *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, 1703–1712. ACM.
- Wiegrefe, S.; and Pinter, Y. 2019. Attention is not not Explanation. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 11–20. Association for Computational Linguistics.