RANCC: Rationalizing Neural Networks via Concept Clustering

Housam Khalifa Bashier Babiker [1], **Mi-Young Kim**[2], **Randy Goebel**[1]
Alberta Machine Intelligence Institute
[1] Department of Computing Science, University of Alberta
[2] Department of Science, Augustana Faculty, University of Alberta
{khalifab, miyoung2, rgoebel}@ulaberta.ca

Abstract

We propose a new self-explainable model for Natural Language Processing (NLP) text classification tasks. Our approach constructs explanations concurrently with the formulation of classification predictions. To do so, we extract a rationale from the text, then use it to predict a concept of interest as the final prediction. We provide three types of explanations: 1) rationale extraction, 2) a measure of feature importance, and 3) clustering of concepts. In addition, we show how our model can be compressed without applying complicated compression techniques. We experimentally demonstrate our explainability approach on a number of well-known text classification datasets.

1 Introduction

Deep neural network (DNN) models provide powerful and sophisticated approaches for addressing Natural Language Processing (NLP) text classification tasks. Yet their underlying behaviour is often opaque, especially in sensitive domains which can critically influence a user's decision, such as in legal and medical domains. As a result, we need to create DNNs that are explainable, in other words, that can provide explanations for their predictions. Our work here focuses on understanding predictions made by a DNN model, to provide explanations at inference time. There has, alternatively, been attention on models attempt to make a neural network explainable, for instance (Sundararajan et al., 2017) and (Ribeiro et al., 2016), which create post-hoc explanations to support explainability. There are also other methods which focused on learning explanations concurrently with the prediction. For example, (Lei et al., 2016) and (Bastings et al., 2019) proposed a neural network architecture for text classification which "justifies" its predictions by selecting relevant tokens in the input text. But this interpretable representation is then adjusted by a complex neural network, so the method is transparent as to what aspect of the input it uses for prediction, but *not* how it captures the salient features. Our work also focuses on extracting a rationale (which can also be defined as an excerpt or justification) simultaneously while computing a prediction. In addition, we insist that explanations are provided interactively, so that a user can switch from one explanation to another to gain a better understanding about the class prediction (sometimes referred to as alternative explanations) as in the work of (Atakishiyev et al., 2020)). Overall, we investigate how to improve the level of abstraction for DNNs and go beyond measuring feature importance. To address this problem, we present a new alternative explanation mechanism, i.e., clustering similar rationales into distinct clusters concurrently with rationale extraction. Our explanation mechanism for DNNs can be summarized as follows:

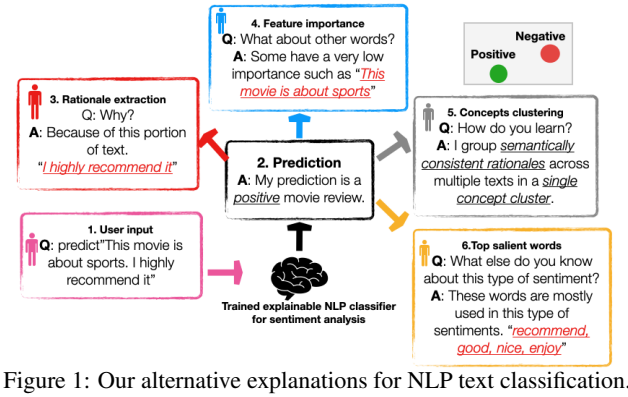


Figure 1: Our alternative explanations for NLP text classification.

Proceedings of the 28th International Conference on Computational Linguistics, pages 3214–3224
Barcelona, Spain (Online), December 8-13, 2020

1) do unsupervised rationale extraction concurrently with classification, 2) dynamically measure feature importance, and 3) learn meaningful concept vectors, i.e., learn meaningful clusters concurrently with rationale extraction.

A concept is represented as a vector in the space which groups all the set of examples (rationales) that share the meaning of the concept. For example, consider sentiment classification that has the following rationales in movie reviews, such as "very nice movie," "enjoyed the movie," "but the acting was very nice," et cetera; all of these text rationales share the same general semantics of "positiveness," and will be represented as a single concept (positive sentiment). Our model learns to group semantically consistent rationales across multiple texts into a single concept cluster. In addition, it is known that a non-linear deep network relies on millions or even billions of parameters, which makes it hard to be deployed in many real-world problems (e.g., on small devices such as cell phones, FPGAs, etc.) due to lack of computation power and the availability of GPUs. We show how our model can be compressed to work in real time applications without sacrificing significant accuracy, and without re-training, parameter pruning, or using quantization techniques. This solution is unique to our model, and to the best of our knowledge, is the first method which combines these techniques in a single model. Figure 1 shows the explainability structure of our model. To explain a prediction, our model can expose the extracted rationale, measure relevant feature importance, and visualize the clustered concept. We can also identify the top salient words based on feature importance.

Our contributions can be summarized as follows: 1) we propose a self-explainable neural network model that concurrently extracts a rationale and predicts the classification output; 2) we extract rationales from input texts, and cluster them into concepts; 3) we improve the explainability of the black-box model by producing the extracted rationales and producing visualized clusters of the constructed concepts; and 4) our model can be compressed without losing much accuracy, and can be deployed as an on-line service in real-time applications in resource-restricted devices.

2 RANCC: Rationalizing Neural Networks via Concept Clustering

We call our model **RANCC** - **R**ationalizing **N**eural **N**etworks via **C**oncept **C**lustering. This section explains the ideas and methods in RANCC: (a) how to build a self-explainable neural network model for text classification (e.g., a model that provides a rationale concurrently with the prediction), (b) how to simultaneously learn concepts of interest from the training data and cluster them, and (c) how to compress RNCC model. The overall model is shown in Figure 2.

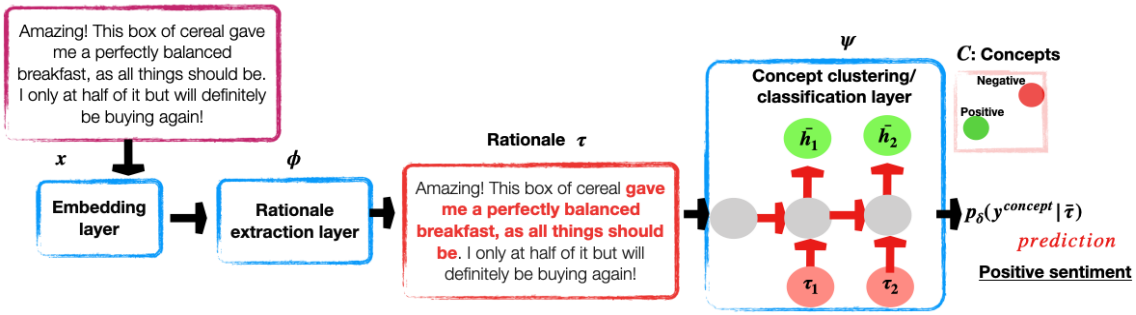


Figure 2: **Block diagram of our method**. A text instance is fed into the embedding layer. The rationale is extracted from the text instance, and pass it to the concept clustering/classification layer to cluster the concept and predict the target class. The loss is computed, and used to update the entire network in an end-to-end approach. Note that the black arrows indicate the steps of our approach and the red arrows indicate the process inside any recurrent network.

2.1 Steps for rationalizing predictions in RANCC

In a text classification task, an input ordered sequence $x = \langle x_1, ..., x_l \rangle$ is mapped to a distribution over class labels using a parameterized θ neural network architecture (such as a Long Short Term Memory network or LSTM), i.e., $\mathcal{F}(x; \theta)$. Normally, the input to $\mathcal{F}$ is in the form of sentences, or short paragraphs. The output y is a vector of class probabilities. The target class $y_i \in y$ is a categorical outcome, such as a sentiment class like "positive review." The distribution over the labels is defined as $y|x \sim Cat(\mathcal{F}(x; \theta))$.

2.1.1 Unsupervised rationale extraction

A rationale is defined as a subset of text extracted from the source document of the task, which provides sufficient evidence for predicting the correct output. Our technique assumes that an explanation of a black-box's prediction is understandable and meaningful if it relies on a small number of words (a rationale), where each rationale relates to parts of text that are semantically consistent across multiple texts. Given an input sequence $x : w_1, ..., w_l$ where w_i, a word in the sentence, is represented as a fixed size vector where $w_i \in \mathbb{R}^d$, and d is the dimension of embedding vectors. For each sequence, we first extract a rationale to be used by the downstream classifier. This rationale is also used as our first explanation to the model's final prediction. We feed x to a function $\phi(x)$ (this function is a convolution operation over the embedding matrix). The function learns v feature maps $\boldsymbol{\mathcal{Z}} = \{\boldsymbol{Z}_i\}_{i=1}^{v}$, whose shape is $l \times d \times v$. For instance, in the case of a movie review input of length 50 words and feature dimension of size 100, the function produces $50 \times 100 \times v$ feature maps. We then aggregated all the feature maps to obtain a single matrix $\bar{\boldsymbol{Z}} \in \mathbb{R}^{l \times d}$. This matrix has the following properties: 1) each row represents a word from the input sentence, 2) if the feature values on a row are larger than average, then the corresponding word has a high probability of selection. We compute the score for each word to be sampled in the rationale as follows :

$$p_{\boldsymbol{\gamma}}(w_1..., w_l|x) = \frac{\exp(\sum_i^l \sum_j^d \bar{\boldsymbol{Z}}_{ij})}{\sum_r^l \exp(\sum_i^l \sum_j^d \bar{\boldsymbol{Z}}_{ij})_r} \tag{1}$$

where r iterates from the first word to the last word. From the probability distribution, we uncover the rationale $\boldsymbol{\tau}$ by sampling $\hat{l}$ words $\boldsymbol{\tau} \sim p_{\boldsymbol{\gamma}}(w_1..., w_l|x)$ which produces the rationale $\boldsymbol{\tau} \in \mathbb{R}^{\hat{l} \times d}$. Note that the length of the rationale $\hat{l}$ is defined by the user. During the test phase, we make predictions based on what is the most likely assignment for each $\boldsymbol{\tau}_i$ using $argmax$. Note that $p_{\boldsymbol{\gamma}}(w_1..., w_l|x)$ can be used to measure the feature importance of each word.

2.1.2 Uncovering concept vectors and class prediction

Our goal is to group $\boldsymbol{\tau}$ into concepts of interest (i.e., to transform rationales into meaningful concept clusters) concurrently with rationale extraction. All rationales belonging to a given concept/class should share common semantics. The main idea of our concept vectors is summarized as follows: 1) every class is represented by a concept vector in the space, 2) each concept vector is used to cluster the rationales into distinct clusters without further training, 3) we predict a concept of interest to present the class label, 4) the concept vectors can be used to create a compressed model, and 5) the visualization of the rationales through the clusters adds another level of abstraction.

Let $\beta \in \mathbb{R}^{\hat{l}}$ denote the probabilities of the sampled words. Given $\boldsymbol{\tau}$, let us suppose that an analyst is interested in a concept representing negative sentiments in movie reviews, and wants to know whether the rationales can be grouped into a single meaningful cluster. The point is that the clustered rationales give the analyst a better understanding on how the model tackles the problem in general, and enables the analyst to understand if the model is able to learn discriminating features from the raw embedding. We first initialize a matrix of weights $\boldsymbol{C} \in \mathbb{R}^{m \times d}$, where m is the number of target concepts and $c_i \in \boldsymbol{C}$ represents the concept vector for the target $y_i \in y$. Our goal is to predict a concept of interest (i.e., a row in the matrix) instead of a class label. β has $\hat{l}$ elements, and $\boldsymbol{\tau}$ has the same $\hat{l}$ rows. We obtain $\bar{\boldsymbol{\tau}}$ through multiplying the values in the i-th row of $\boldsymbol{\tau}$ by β_i. We feed the rationale to any learning function, e.g., an LSTM ψ i.e., $\psi(\bar{\boldsymbol{\tau}})$ to obtain a new latent representation $\hat{\boldsymbol{H}} \in \mathbb{R}^{\hat{l} \times d}, \hat{h}_i \in \hat{\boldsymbol{H}}$. The last state $\hat{h}_{\hat{l}}$ is fed into a non-linear layer with parameters $\boldsymbol{\delta}_{concept} \in \mathbb{R}^{d \times m}$ which produces a score for every concept as follows:

$$p_{\boldsymbol{\delta}}(y^{concept}|\bar{\boldsymbol{\tau}}) = \frac{\exp(\hat{h}_{\hat{l}} \cdot \boldsymbol{\delta}_{concept})}{\sum_{k=1}^{m} \exp(\hat{h}_{\hat{l}} \cdot \boldsymbol{\delta}_{concept})_k} \tag{2}$$

where, $y^{concept}$ is the output probabilities of shape $1 \times m$. Using this distribution we can predict a specific concept : $y_s^{concept} \sim p_{\boldsymbol{\delta}}(y^{concept}|\bar{\boldsymbol{\tau}})$. Using $y_s^{concept}$, we can extract c_s, which is the corresponding

concept vector of the detected concept. The role of the concept predictor is to further enforce concept consistency, i.e., the accurate prediction of a class is subject to consistent concepts. This allows concepts to be jointly trained with the rest of the model. The predicted concept class should match the target ground-truth label.

2.2 Formulating the learning problem in RANCC and compressing RANCC

Our objective function aims to learn the following tasks: (a) learning a rationale from text input, (b) predicting a concept of interest, and (c) clustering a rationale based on its concept.

2.2.1 Learning rationales

Our loss function for the rational extraction aims to maximize the probabilities of salient words from the rationale extractor layer. In general, the loss maximizes the log probability of the selected words that lead to a correct prediction. Our optimization objective is defined as follows:

$$\mathcal{L}^{rationale} = \lambda\Bigg(-\sum_{i}^{s} A_i \log p(\beta|x) \Bigg), \tag{3}$$

where β is the probabilities of the selected words, s is the batch-size, λ is used to weight the importance of this loss, and A_i is a scalar. For example, a scalar A_i could be 1 if the model predicts the correct class label for x using the rationale τ and 0 otherwise. We used a custom gradient to pass the updates through the sampling step in the rationale extraction layer. The custom gradient function works as follows: first assign a gradient of 1 to each selected $\boldsymbol{x}_i$, and 0 otherwise.

2.2.2 Learning concept vectors

Each concept vector should correspond to semantically consistent rationales, i.e., all the rationales belonging to a given concept should share common semantics. We assume that every class has only one single concept of interest. Each rationale must be assigned to a single concept vector and here we describe a way of integrating a clustering technique within a neural network text classification, for grouping rationales. Our approach is capable of capturing the local structure of the high-dimensional data simultaneously with other tasks. We use cosine distance to cluster rationales based on their concepts, i.e., grouping similar rationales into a single cluster. The cosine distance is computed as follows:

$$\mathcal{L}^{clusters} = \begin{cases} 1 - \frac{\hat{\tau} \cdot c_s}{\|\hat{\tau}\| \|\mathbf{c}_s\|}, & \text{if } s = y_i \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\cdot$ is the dot product operation, $\hat{\boldsymbol{\tau}}$ is obtained by taking the average over the columns of $\bar{\boldsymbol{\tau}}$ and s is the index of the predicted concept vector c_s. This loss is only applied if the prediction at the output layer is correct given the rationale $\boldsymbol{\tau}$ and the concept vector c_s, otherwise the loss is set to zero. The reason for doing this is to minimize the cosine distance only between the correct concept vector and rationale (i.e., if extracted rationale resulted in a correct class prediction). By doing so, we are grouping similar rationales into a single concept. Every rationale $\boldsymbol{\tau}$ is clustered around its concept by computing the mean (x-axis) and the standard deviation (y-axis) of $(c_s + \hat{\boldsymbol{\tau}})$ from every rationale.

2.2.3 Classifying rationales based on concepts

For text classification, we use the standard cross-entropy loss function to penalize miss-classifications based on the predicted concept vector:

$$\mathcal{L}^{classify} = -\frac{1}{m}\sum_{i=1}^{m} y_i \log(y_i^{concept}), \tag{5}$$

where $y_i^{concept}$ is the predicted probability for class y_i. We jointly learn the rationale extractor and concept predictor. The final objective function is the sum of Equations (5), (3) and (4).

2.2.4 Model compression without extra computation

To create a compressed model, we create a simple classifier only using the embedding features and the concept vector of each class. The classifier is based on the cosine similarity between the average of the raw embedding features of the input sentence and the concept vector of each class. The target class is determined as the corresponding label of the concept vector whose cosine similarity is the highest. We will show the performance of the compressed model in Section 4. The summary of this solution is shown in Figure 3.

Input: Sentence average embedding t, Concepts $\boldsymbol{C}$
Output: Target concept class $\bar{y}$
for Each concept $c_i \in \boldsymbol{C}$ **do**
  $tmpSim$ = Similarity (c_i, t)
  if $tmpSim > sim$ **then**
    $\bar{y} \leftarrow i$
    $sim \leftarrow tmpSim$
  end if
end for
Figure 3: Compressed model algorithm.

3 Related work

Most of the emerging work on explainability relies on approximating feature importance from a pre-trained network, typically using perturbation and gradient-based methods (Sundararajan et al., 2017; Shrikumar et al., 2017a; Arras et al., 2017; Zeiler and Fergus, 2014). These methods assign a score to each word in the input text w.r.t the predicted class. One of the problems with post-hoc explanation is that it is often computationally expensive. For instance, the method proposed by (Zintgraf et al., 2017) takes up to one hour to produce a result. However, our approach is different, as it attempts to build a self-explainable network. There are other post-hoc techniques that relied on decomposing the output of LSTMs to learn feature importance (Murdoch and Szlam, 2017). They introduce a technique which decomposes the output of the LSTM into a sum over word coefficients, then show how these coefficients are meaningful with respect to a prediction — they later improved the approach (Murdoch et al., 2018). Several other methods are based on learning from word vectors such as (Faruqui et al., 2015), while some used auto-encoders with graph theory to provide explanations (Alvarez-Melis and Jaakkola, 2017). Another approach for tackling explainability in text classification is to learn a rationale, i.e., a subset of short and meaningful features from the text input. These methods focus on augmenting the neural network with another network architecture to uncover the rationale by finding out which portions of the input contribute most to the prediction of a target class (Lei et al., 2016; Bastings et al., 2019). Their neural network creates a Bernouli distribution over the set of the input variables. During training they sample from the rationale-network and during inference they use $argmax$ over the distribution. However, the level of the abstraction of these methods is limited to the input level. In our case, we create a self-explainable neural network to extract the rationale, thus making our approach end-to-end learning, i.e., we only use a single neural network to extract a rationale for explainability. In addition, we go beyond the rationales by introducing concepts and clustering rationales through concepts without additional computation. Recently some work started to consider concept extraction for NLP (Bouchacourt and Denoyer, 2019). However, our approach attempts to learn meaningful clusters for concepts while the existing work attempts to extract individual words to represent the concepts. Finally, there is a debate on whether attention can be used for explanation or not. This is beyond the scope of this work and we refer the reader to the following papers (Serrano and Smith, 2019) and (Jain and Wallace, 2019). In addition, our work is different from topic modelling methods (Blei et al., 2003) as we attempt to learn clusters as well as rationales concurrently with the classification in neural networks.

4 Experiments

Our primary intent is *not* predictive accuracy. We used standard practices for training without much tuning. The aim of the experiments can be summarized as follows: 1) we show how RANCC outperforms the state-of-the-art baselines for rationale extraction, 2) we show how RANCC outperforms feature importance methods (i.e., post-hoc explanation methods), 3) we show how RANCC can be used to cluster rationales without using any dimensionality reduction technique, 4) we visualize the concepts learned by the classifier, 5) we visualize a few samples of the extracted rationales, and 6) we show the performance of the compressed RANCC model. The hyperparameters used for the experiments are shown in Table 1. Our implementation is available here. [1]

[1] https://github.com/housamkhalifa/rancc

Optimizer	Adam
Text length	50 words for IMDB, 20 words for AGnews
Learning rate	$1e-3$
Embedding dimension	300
Concept vector dimension	300
Cell	LSTM
LSTM Hidden dimensions	300
Scalar A	1.0 if correct class prediction, 0.0 otherwise
Batch-size	256

Table 1: Hyperparameters used in the experiments.

4.1 Evaluating our method against self-explainable models

Objective. When rationalizing neural network text classification prediction, our goal is to perform as well as systems using the full input text, while using only a subset of the text, leaving unnecessary words out for explainability.

Rationalizing text prediction in sentiment analysis. We use the IMDB dataset that was proposed by (Maas et al., 2011) for sentiment classification with two labels, positive and negative sentiments. It consists of $25,000$ instances for training and $25,000$ instances for testing. We compare our approach against the self-explainable approach proposed by (Bastings et al., 2019). For fair comparison, we followed the hyperprameter suggested by the authors for implementing the baseline. We followed the metric proposed by (Bastings et al., 2019), and we computed the accuracy as a function of the length of the extracted rationale. The higher the accuracy, the better the approach. Figure 4 shows the performance for various percentages of selected text. Our approach outperforms the work of (Bastings et al., 2019), achieving a similar accuracy as the baseline system which uses the full text, by using only the top 10% words in the text. In addition, our model showed better accuracy than (Bastings et al., 2019) throughout all the selected text experiments. This shows that RANCC captures better discriminating features than the previous self-explainable model.

Rationalizing text prediction in news classification. We use the AGnews dataset (Zhang et al., 2015) to test the performance on topic classification. The dataset consists of 127600 samples divided into 4 classes. We split it into training set 80% and testing set 20%. Figure 4 shows the results on AGnews, and we can observe that our approach outperforms the baseline and the work of (Bastings et al., 2019), achieving better performance.

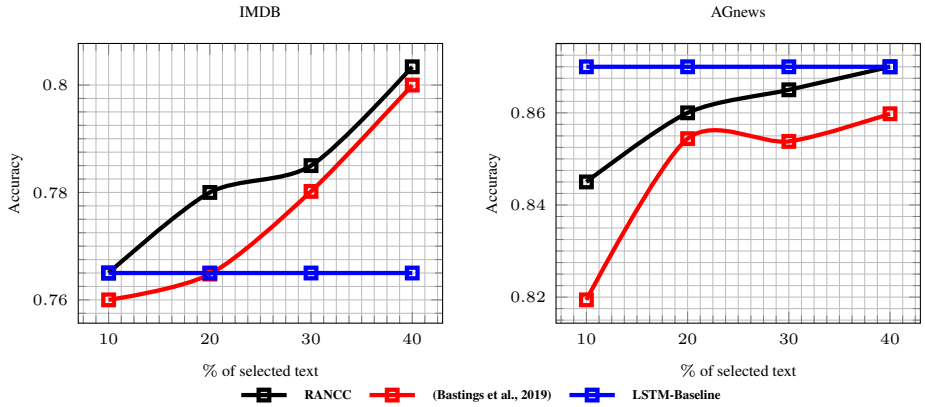


Figure 4: IMDB test accuracy (left) and AGnews test accuracy (right) for various percentages of extracted text. Baseline refers to the LSTM network trained on the full text.

4.2 Faithfulness: are "relevant" features truly faithful to what the model computes?

Goal. Verify whether the estimated feature importance is actually "faithful" to what the model actually computes. Common techniques for evaluating the importance rely on observing the effect on model's

prediction after removing a meaningful feature. In this subsection, we evaluate the faithfulness against post-hoc explanations, by comparing the feature importance approximated by our approach and that of the post-hoc methods. We use the AGnews dataset and IMDB, and the data is divided into the same training and test sets of the previous experiments. We compare our method with several competitive algorithms for feature importance scoring on black-box models, including gradient-based methods such as ϵ-**LRP** (Bach et al., 2015), **Grad*Input** (Shrikumar et al., 2017b), and **Intgrad** (Sundararajan et al., 2017) . For perturbation-based methods, we compare our approach against **LIME** (Ribeiro et al., 2016).

Change in log-odds ratio. The objective is to determine if the mean log-odds ratio of the predicted class decreases as the percentage of masked features over the total number of features increases. We mask the top k features ranked by importance score and those masked words are replaced by zero padding. We then feed the input and measure the drop of the values between the probabilities of the target class when no word is deleted, and when k words are removed. Figure 5 shows the results of the change in log-odds ratio experiment on the AGnews dataset. Note that **Grad*Input** has the same performance as ϵ-**LRP**. We can see that our method achieves the lowest log-odds ratio (the biggest change in log-odds ratio) when removing salient features from the text input. This could be because our approach is not post-hoc and thus the optimization considers maintaining both of the accuracy and explainability. Therefore, RANCC could correctly capture the important features affecting the prediction output. From our results, we can conclude that the explanation method produces a faithful explanation about which factors were important in that calculation, so we can consider the explanation to be faithful to the model. This metric was used in literature for model's explanations (Shrikumar et al., 2017a).

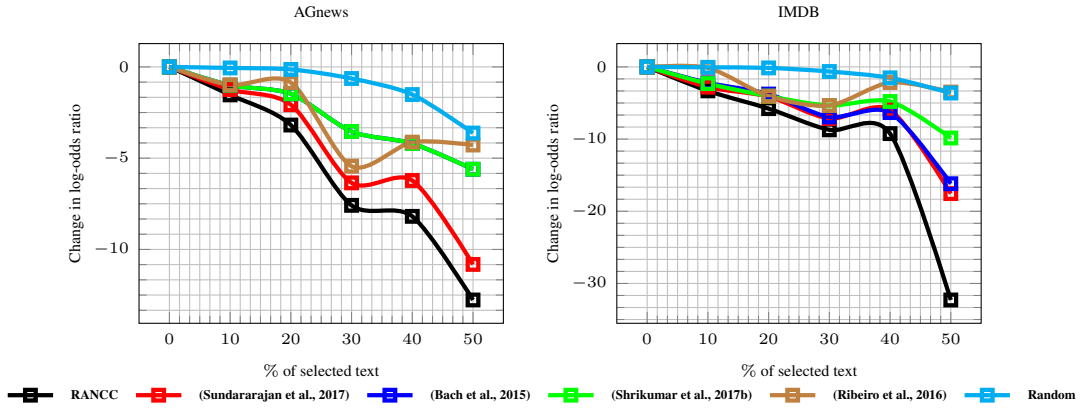


Figure 5: Change of log-odds ratio for various percentages of extracted rationale. Lower log-odds scores are better.

4.3 Jensen-Shannon divergence of model output distributions

We compare the impact of the model's computed distribution when no feature is removed and when k features are removed. Our approach here is to calculate the Jensen-Shannon Divergence (JSD) between the two distributions. We show the results on the post-hoc explanation techniques in Figure 6. This experimental result shows that the removed features within RANCC affect the output distribution of the model much more than the other methods. This means that RANCC captured the important features better than the other methods. Given the countless number of baselines and rapid development in this area, we note that our performance is not compared against all of the existing methods, largely due to challenges with source code access, and space limitations for this document.

4.4 Visualizing concept clusters

Our approach is also capable of clustering data points simultaneously with the classification task, i.e., our neural network is capable of providing distinct clusters related to specific concepts. The clustering results on IMDB are shown in Figure 7. As we can observe, our approach achieves better clusters than t-SNE (Maaten and Hinton, 2008) and PCA. Our method groups movie reviews into unique clusters based on their concept vectors without using dimensionality reduction techniques. Figure 7 shows that our concepts of positive and negative sentiments are clearly separated from each other. The clustering

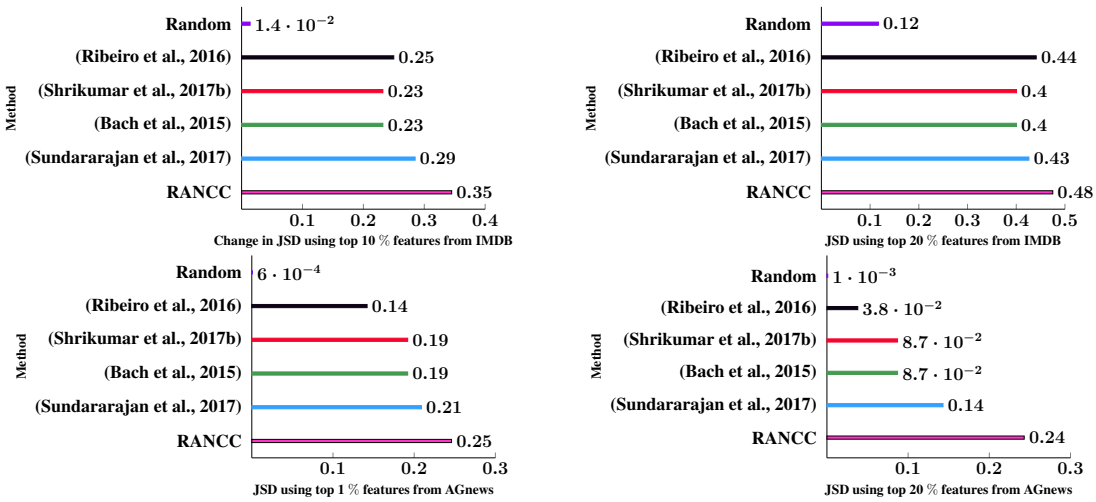
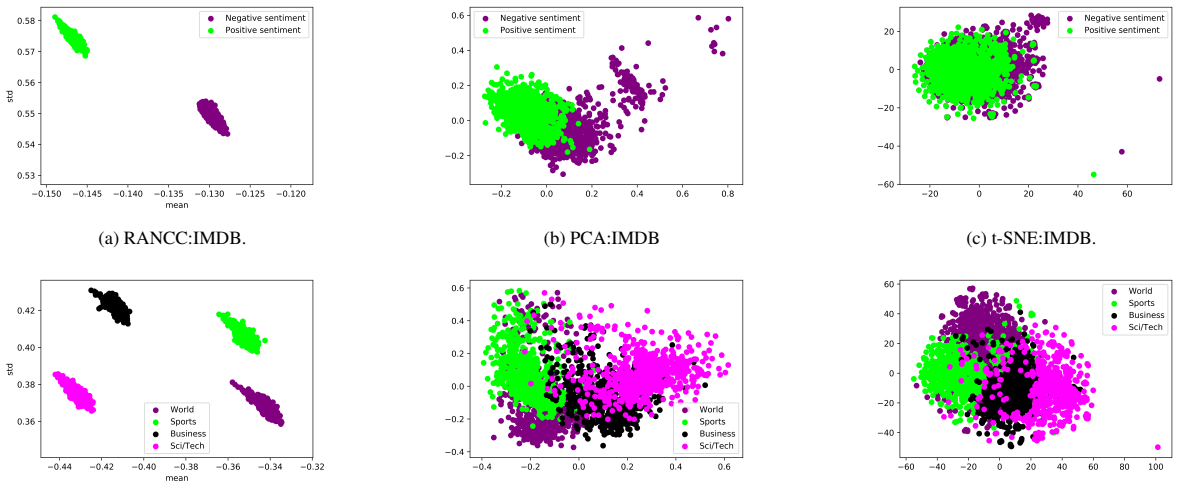


Figure 6: JSD on the IMDB and AGnews. The higher the value, the better the method.

results on AGnews are also shown in Figure 7 which displays the 4 clusters (these are the four classes in the dataset representing four concepts). Our work does a better job of revealing the natural classes in the data than t-SNE and PCA, and thus RANCC is better in accurately generating distribution and partitioning the data. The idea of concurrently clustering/grouping rationales into clusters is a unique feature in our approach which has been learned without additional computation. The merit is that, you do not need to use a clustering algorithm over the embedding to obtain the target cluster. This clustering helps explain the extracted rationales and prediction outcomes. This approach makes our work unique compared to other explainability methods. Additionally, to the best of our knowledge, this is the first approach to provide concepts in terms of clusters and to be able to learn them simultaneously for text classification in a deep learning model.



(a) RANCC:IMDB. (b) PCA:IMDB (c) t-SNE:IMDB.

(d) RANCC: AGnews. (e) PCA: AGnews (f) t-SNE:AGnews.

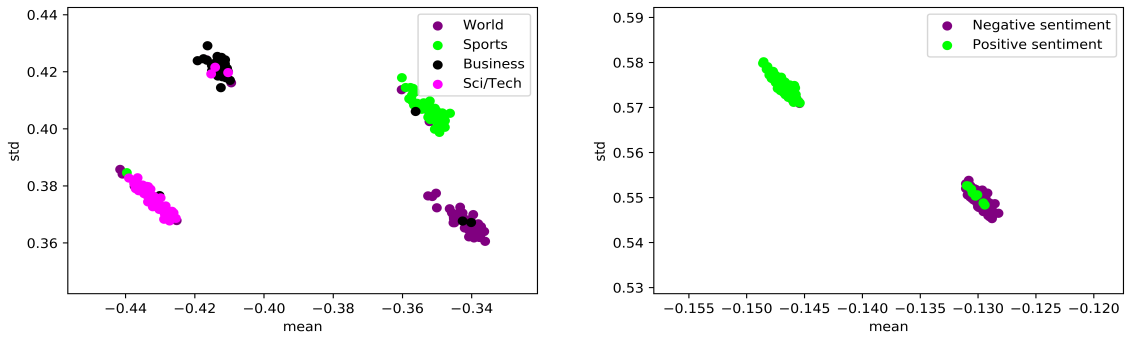
Figure 7: Clusters of correctly classified rationales using concept vectors for both IMDB and AGnews using RANCC, PCA and t-SNE. Please note that the mean and standard deviation are obtained from the clustering.

.

4.5 Using the clustering visualization as a debugging tool

One of the challenges when debugging a neural network model is to understand errors and when they occur. More specifically, we are interested in which class the model mistakenly classified the text input. A general solution is to write a piece of code and use many print statements to debug it and to understand the miss-classified input. But by using the clusters, we can easily visualize the miss-classified inputs and understand to which class the input was incorrectly classified. Looking at the results from Figure 8, one

can visually identify the errors and can save much time for debugging, rather than using print statements or other clumsy alternatives. For instance, Figure 8 shows that in the AGnews, the model struggles in learning meaningful features for the business class and the same issue is applied to the IMDB for the positive sentiment.



(a) RANCC: Debugging AGnews. (b) RANCC:Debugging IMDB.

Figure 8: Clusters of rationales for identifying miss-classification.

4.6 Understanding concepts

An important question is what kinds of words are encoded in each rationale to represent a concept of interest? Each discovered concept can be understood from the corresponding rationale which activates its appearance: for example in the AGnews, the concept the model identifies is inferred from the following detected word: *Microsoft*, *Software*, and *Internet*. With these words, we can identify that the concept of these texts is "science/technology." The same reasoning applies to the other rationales such as *Company*, *Inc, Oil,* and *Price* which correspond to the "business" concept. The other rationales, such as *President, government, leader,* and *official*, correspond to the "world news" concept, and *game*, and *team* correspond to the "sport news" concept (See Table 2). The same reasoning can be applied to the IMDB movie reviews. Our model is able to discover interesting words which correspond to meaningful concepts as shown in Table 3.

Class	Top words
World news	president, government, leader, official, security, war, attack, nation, police, foreign
Sport news	sport, team, victory, football, final, home, season, game,club, fan, champion, player, title, championship, star, field
Tech/Sci news	technology, program, microsoft, software, internet, service, window, network, pc, operating system
Business news	market, company, billion, firm, cost, cut, federal, report,profit, earning, research, international,share

Table 2: Visualizing top words used in each concept for AGnews.

Class	Top words
Positive sentiment	interesting, pretty, original, horror, cool, kind,great, see
Negative sentiment	bad, waste, worst, poor, boring, stupid, awful

Table 3: Visualizing top words used in each concept for IMDB.

4.7 Visualizing extracted rationales

We also visualize the rationales extracted from the IMDB movie reviews on a single concept from IMDB (i.e., positive sentiment concept). As we can see from Figure 9, our approach is capable of capturing meaningful rationales. Through the highlighted rationales, we can see that they are semantically similar, that is, they can be grouped in a single consistent concept.

4.8 Model compression

We show how our compressed model achieves close results to the original trained RANCC model, without re-training. We show the performance on the AGnews dataset (see Table 4). In Table 4, 10% and

hour of the film had me in tears with the honesty of the emotions is not everyone's cup of tea but unlike the little she has written some truly sympathetic wonderful characters and a fine story given a casting and production values by warner brothers highly recommended
stumbles on to three other strange with past residents of the same house i won't say anymore for i will ruin the movie more than i already have but it is a terrific movie for as old as it is and would never mind watching it again
comedy which is a very fitting title for this movie as well as for the whole genre that practically invented and the cult favorite for over 20 years 1986 is brilliant and deserves our true love and genuine for the unforgettable moments of cinematic happiness
a great selection of the finest british talent around i loved them all for every diverse element brought into the film italy has to be one of the most romantic places to form a story such as this everything about this film works i love it
che's capture and death are dealt with well the film is greatly enhanced by the dialogue being in spanish del toro is again excellent as the charismatic so if you've seen part 1 you will see a very similar telling of a very different story in part 2.

Figure 9: Visualizing the extracted rationales of different lengths. Highlighted text represents the extracted rationale for the positive concept.

40% mean the percentages of the extracted words as a rationale. Our compressed model shows close performance to the original RANCC model and without much loss in performance.

Method	**F1**	**Recall**	**Precision**	**Accuracy**
RANCC (10%)	0.856	0.857	0.858	0.86
Compressed (10%)	0.827	0.857	0.858	0.827
RANCC (40%)	0.876	0.876	0.878	0.87
Compressed (40%)	0.865	0.866	0.866	0.866

Table 4: Comparison of the performances between RANCC and compressed RANCC.

5 Conclusion and Future work

We have presented a new approach for a self-explainable neural network applied to text classification, and presented new techniques for alternative explanations. Our extracted rationals were important features affecting the prediction, and the visualization of the concept clustering improved the explainability of black-box models. In future work, we are interested in understanding more about the semantic distance between rationales and the concept vectors, as well as the semantic distance between every word in a sequence and its concept vector. We will also further investigate the explainability of the compressed RANCC model, and aim to investigate our approach's explainability in other NLP tasks, such as natural language inference, and language generation in a specific domain of medicine or law.

Acknowledgements

We acknowledge support from the Alberta Machine Intelligence Institute (AMII), from the Computing Science Department of the University of Alberta, and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

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

Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis in ACL*, pages 159–168.

Shahin Atakishiyev, Housam Babiker, Nawshad Farruque, Randy Goebel, Mi-Young Kim, Mohammad Hossein Motallebi, Juliano Rabelo, Talat Syed, and Osmar R. Zaïane. 2020. A multi-component framework for the analysis and design of explainable artificial intelligence. *arXiv preprint arXiv:2005.01908*.

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.

Joost Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of ACL*, pages 2963–2977.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Diane Bouchacourt and Ludovic Denoyer. 2019. Educe: Explaining model decisions through unsupervised concepts extraction. *arXiv preprint arXiv:1905.11852*.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith. 2015. Sparse overcomplete word vector representations. In *Proceedings of ACL*, pages 1491–1500.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *Proceedings of EMNLP-IJCNLP*, pages 3543–3556.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of EMNLP*, pages 107–117.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL*, pages 142–150. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

W James Murdoch and Arthur Szlam. 2017. Automatic rule extraction from long short term memory networks. In *Proceedings of International Conference on Learning Representation (ICLR)*.

W James Murdoch, Peter J Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from lstms. In *Proceedings of the International Conference on Learning Representation (ICLR)*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017a. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017b. Learning important features through propagating activation differences. In *Proceedings of the International Conference on Machine Learning*, pages 3145–3153.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of International Conference on Machine Learning (ICML)*, page 3319–3328.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision*, pages 818–833. Springer.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.

Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. 2017. Visualizing deep neural network decisions: Prediction difference analysis. In *Proceedings of ICLR*.