Hierarchical Interpretation of Neural Text Classification

Hanqi Yan*
Department of Computer Science
University of Warwick, UK
Hanqi.Yan@warwick.ac.uk

Lin Gui*
Department of Informatics
King's College London, UK
lin.1.gui@kcl.ac.uk

Yulan He
Department of Informatics
King's College London, UK
University of Warwick, UK
The Alan Turing Institute, UK
yulan.he@kcl.ac.uk

Recent years have witnessed increasing interest in developing interpretable models in Natural Language Processing (NLP). Most existing models aim at identifying input features such as words or phrases important for model predictions. Neural models developed in NLP, however, often compose word semantics in a hierarchical manner. As such, interpretation by words or phrases only cannot faithfully explain model decisions in text classification. This article proposes a novel Hierarchical Interpretable Neural Text classifier, called HINT*, which can automatically generate explanations of model predictions in the form of label-associated topics in a hierarchical manner. Model interpretation is no longer at the word level, but built on topics as the basic semantic unit. Experimental results on both review datasets and news datasets show that our proposed approach achieves text classification results on par with existing state-of-the-art text classifiers, and generates interpretations more faithful to model predictions and better understood by humans than other interpretable neural text classifiers.*[1]

* Equal contribution.

[1] Our source code can be accessed at https://github.com/hanqi-qi/SINE.

Action Editor: Tal Linzen. Submission received: 10 August 2021; revised version received: 17 July 2022; accepted for publication: 3 August 2022.

https://doi.org/10.1162/coli_a_00459

1. Introduction

Deep Learning (DL) models have achieved state-of-the-art performance in many NLP tasks (Devlin et al. 2019; Yang et al. 2019; Brown et al. 2020; Yan et al. 2021). A deep neural network containing many layers is usually viewed as a black box that has limited interpretability. Recently, the field of explainable AI has exploded with various new approaches proposed to address the problem of the lack of interpretability of deep learning models (Lipton 2018; Jacovi and Goldberg 2020; Ribeiro et al. 2020).

Methods for the interpretation of DL models can be broadly classified into post-hoc interpretation methods and self-explanatory methods. The former typically aims to establish the relationship between the changes in the prediction output and the changes in the input of a DL model in order to identify features important for model decisions. For example, Jawahar, Sagot, and Seddah (2019) used probing to examine BERT intermediate layers. Abdou et al. (2020) modified input text by linguistic perturbations and observed their impacts on model outputs. Selvaraju et al. (2020) tracked the impact from gradient changes. Kim et al. (2020) erased word tokens from input text by marginalizing out the tokens. On the other hand, the self-explanatory models are able to generate explanations during model training by "twinning" a black-box Machine Learning model with transparent modules. For example, in parallel to model learning, an addition module is trained to interpret model behavior and is used to regularize the model for interpretability (Alvarez-Melis and Jaakkola 2018; Rieger et al. 2020). Such models, however, usually require expert prior knowledge or annotated data to guide the learning of interpretability modules. Chen and Ji (2020) proposed to improve the interpretability of neural text classifiers by inserting variational word masks into the classifier after the word embedding layer in order to filter out noisy word-level features. The interpretations generated by their model are only at the word-level and ignore hierarchical semantic compositions in text.

We argue that existing word-level or phrase-level interpretations are not sufficient for interpreting text classifier behaviors, as documents tend to exhibit topic and label shifts. It is therefore more desirable to explore hierarchical structures to capture semantic shifts in text at different granularity levels (O'Hare et al. 2009), Lin et al. 2012; Yang et al. 2016; Wang et al. 2020; Arnold et al. 2019; Xie et al. 2021; Gui and He 2021). Moreover, simply establishing the relationship between the changes in the input and the changes in the output in a DL model could identify features that are important for predictions, but ignores subtle interactions among input features. Recent approaches have been developed to build explanations through detecting feature interactions (Singh, Murdoch, and Yu 2019; Chen, Zheng, and Ji 2020; Jin et al. 2020; Gui et al. 2022). Nevertheless, they are only able to identify sub-text spans that are important for model decisions and largely focus on sentence-level classification tasks.

We speculate that a good interpretation model for text classification should be able to identify the key latent semantic factors and their interactions that contribute to the model's final decision. This is often beyond what word-level interpretations could capture. To this end, considering the hierarchical structure of input documents, we propose a novel Hierarchical Interpretable Neural Text classifier, called HINT, which can generate interpretations in a hierarchical manner. One example output generated by HINT is shown in Figure 1 in which a review document consisting of 6 sentences (shown in the upper box) is fed to a classifier for the prediction of a sentiment label. Traditional interpretation methods can only identify words that are indicative of sentiment categories, as shown in the middle left box in Figure 1. However, it is still unclear how these words contribute to the document-level sentiment label, especially

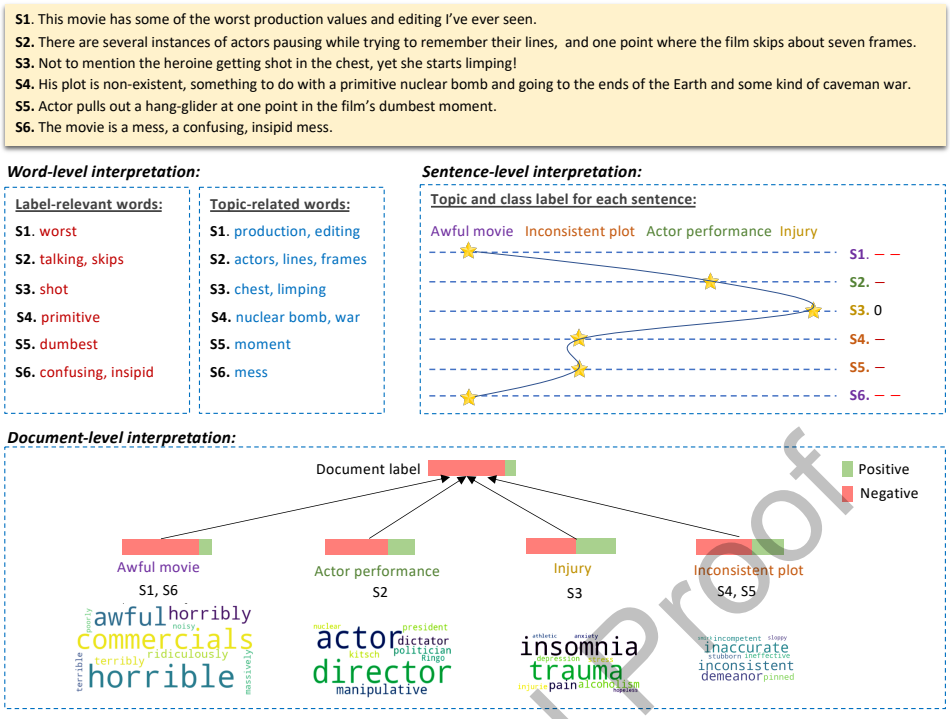


Figure 1
A hierarchical explanation example generated by our proposed method, HINT, on an IMDB review. **Upper**: A review document. **Middle**: The left part shows the word-level interpretation including the label-relevant and the topic-related words, while the right part shows the sentence-level interpretation where the topic and class label for each sentence is presented, for example, Sentence 4 is about the topic "*Inconsistent plot*" with a *negative* polarity. **Lower**: The document-level interpretation shows the topic partition for the 6 sentences in the review. The size of a word cloud indicates its relative importance in the document. The red and green color bar above each word cloud indicates the sentiment of the corresponding topic. It is clear that the negative polarity of the review is mainly due to the general negative comments of *awful movie* and more concretely the complaint about the *actors' performance* and the *inconsistent plot*. The word-, sentence- and document-level interpretations are generated automatically by our model.

when there are words with mixed polarities. Moreover, humans may also be interested in topics discussed in the document and their associated sentiments, and how they are combined to reach the final document-level class label. The middle center box highlights the important topic words in each sentence. Note that traditional post hoc word-level interpretation methods would not be able to identify these words since they are less relevant to sentiment class labels when considered in isolation. The middle right box in Figure 1 shows the associated topic for each sentence and its respective sentiment label. For example, the sentence *S*2 is associated with the topic "*Actor performance*" and has a *negative* polarity. The lower box shows the topic partition of sentences based on their topic semantic similarities. Such hierarchical explanations (from word-level label-dependent and label-independent interpretations, to sentence-level topics with their associated labels, and finally to the document-level topic partition) are generated automatically from our proposed approach. The only supervision information required for model learning are documents paired with their class labels.

As will be shown in the Experiments section, our proposed approach achieves comparable classification performance compared to the existing state-of-the-art neural text classifiers when evaluated on three document classification datasets. Moreover, it generates interpretations more faithful to model predictions and better understood by humans compared to word-level interpretation methods. In summary, our contributions are 3-fold:

- We propose a neural text classifier with built-in interpretability that can generate hierarchical explanations by identifying both label-dependent and topic-related words at the word-level, detecting topics and their associated labels at the sentence-level, and finally producing the document-level topic and sentiment composition.
- The evaluation of explanations generated by our approach shows that it generates interpretations better understood by humans and more faithfully for model predictions compared to existing word-level interpretation methods.
- Experimental results show that our proposed approach performs on par with the existing state-of-the-art methods on the three document classification datasets.

2. Related Work

Our work is related to the following lines of research:

Post-hoc Interpretation. Post-hoc interpretation methods typically aim to identify the contribution of input attributes or features to model predictions. For example, Wu et al. (2020) proposed a perturbation-based method to interpret pre-trained language models used for dependency parsing. Niu et al. (2020) evaluated the robustness and interpretability of neural-network-based machine translation models by slightly perturbing input. Abdou et al. (2020) proposed a new dataset for evaluating model interpretability by seven different types of linguistic perturbations. Kim et al. (2020) argued that interpretation methods that measure the changes in prediction probabilities by erasing word tokens from input text may face the out-of-distribution (OOD) problem. They proposed to marginalize out a token in an input sentence to mitigate the OOD problem. Jin et al. (2020) and Chen, Zheng, and Ji (2020) built hierarchical explanations through detecting feature interactions. But their models can only identify sub-text spans that are important for model decisions and largely focus on sentence-level classification tasks.

Self-Explanatory Models. Different from post-hoc interpretation, self-explanatory methods aim to generate explanations during model training with the interpretability naturally built in. Existing work utilizes mutual information (Chen et al. 2018; Guan et al. 2019), attention signals (Zhou, Zhang, and Yang 2020), Bayesian network (Chen et al. 2020; Tang, Hahn-Powell, and Surdeanu 2020), or information bottleneck (Alvarez-Melis and Jaakkola 2018; Bang et al. 2021) to identify the key attributes or features from input data. For example, Zhou, Zhang, and Yang (2020) used a variational autoencoder based classifier to identify operational risk in model training. Chen et al. (2020) recognized name entities from clinical records and used a Bayesian network

to obtain interpretable predictions. Zhang et al. (2020) proposed an interpretable relation recognition approach by Bayesian Structure Learning. Tang, Hahn-Powell, and Surdeanu (2020) proposed a rule-based decoder to generate rules for model explanation. Zanzotto et al. (2020) proposed a kernel-based encoder for the interpretable embedding metric to visualize how syntax is used in inference. Jiang et al. (2020) incorporated regular expressions into recurrent neural network training for cold-start scenarios in order to obtain interpretable outputs. Chen and Ji (2020) proposed variational word masks (VMASK) that are inserted into a neural text classifier after the word embedding layer in order to filter out noisy word-level features, forcing the classifier to focus on important features to make predictions.

In general, existing self-explanatory methods mainly focus on tracking the influence of input features on model outputs and use it as constraints for model learning. But they ignore the subtle interplay of input attributes. In this article, we propose a novel hierarchical interpretation model, which can generate interpretations at different granularity levels and achieve classification performance on par with the existing state-of-the-art neural classifiers.

Interpretation Based on Attentions. The attention mechanisms have been widely used in neural architectures applied to various NLP tasks. It is common to use attention weights to interpret models' predictive decisions (Li, Monroe, and Jurafsky 2016; Lai and Tan 2019; De-Arteaga et al. 2019). In recent years, however, there has been work showing that attention is not a valid explanation. For example, Jain and Wallace (2019) found that it is possible to identify alternative attention weights after the model is trained, which produced the same predictions. Serrano and Smith (2019) modified attention weights in already-trained text classification models and analyzed the resulting differences in their predictions. They concluded that attention cannot be used as a valid indicator for model predictions. While the aforementioned work modified attention weights in a post-hoc manner after a model was trained, Pruthi et al. (2020) proposed modifying attention weights during model learning and produced models whose *actual* weights could lead to deceived interpretations. Wiegreffe and Pinter (2019) argued the validity of the claim in prior work (Jain and Wallace 2019) and proposed alternative experimental design to test when/whether attention can be used as explanation. Their results showed that prior work does not disprove the usefulness of attention mechanisms for explainability.

3. Hierarchical Interpretable Neural Text Classifier (HINT)

Our proposed **H**ierarchical **I**nterpretable **N**eural **T**ext (HINT) classification model is shown in Figure 2. For each sentence in an input document, a dual representation learning module (§3.1) is used to generate the contextual representation guided by the document class label and the latent topic representation, from which the word-level interpretations can be generated. To aggregate the sentences with similar topic representations, we create a fully connected graph (§3.2) whose nodes are initialized by the sentence contextual representations and edge weights are topic similarity values of the respective sentence nodes. Sentence interactions are captured by a single-layer Graph Attention Network to derive the document representation for classification. In what follows, we describe each of the modules of HINT in detail. The notations used in this article are shown in Table 1.

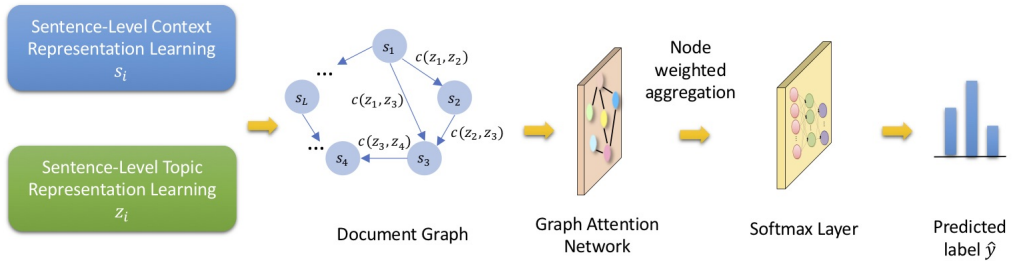


Figure 2
Overall architecture of our proposed dual representation learning framework, which consists of two main modules: (a) for a given sentence i, the *sentence-level context representation learning* module generates the context representation $\boldsymbol{s}_i$ while the *sentence-level topic representation learning* module produces the topic representation $\boldsymbol{z}_i$; (b) A document graph is constructed in which each node represents a sentence and its embedding is initialized by its associated context representation s_i, the weight of the edge connecting two nodes is determined by the similarity between their corresponding topic representations, denoted as $c(\boldsymbol{z}_i, \boldsymbol{z}_j)$. Node representations are updated by a Graph Attention Network and the document representation is derived by weighted aggregation of the node representations. Finally, the document representation is fed to a softmax layer to predict the class label $\hat{y}$.

3.1 Dual Module for Sentence-Level Representation Learning

The dual module captures the sentence contextual and latent topic information separately. In particular, we hoped that the sentence-level context representation would capture the label-dependent semantic information, while the sentence-level topic representation would encode label-independent semantic information shared across documents regardless of their class labels.

3.1.1 Context Representation Learning. In the **sentence-level context representation learning module** shown in Figure 3a, the goal is to capture the contextual representation of a sentence with word-level label-relevant features. We choose a bidirectional LSTM (biLSTM) network, which captures the contextual semantics information conveyed in a sentence, with an attention mechanism that can capture the task relevant weights for interpretation (Yang et al. 2016).

Assuming that a document d contains M_d sentences, $\boldsymbol{w}_d = \{s_1, s_2, ...s_{M_d}\}$, and each sentence indexed by i contains L words with each word represented by a pre-trained N-dimensional word embedding, $\boldsymbol{x}_i = \{x_{i1}, x_{i2}, ...x_{iL}\}, x_{ij} \in \mathbb{R}^N$, then the hidden representation for each word x_{ij}, denoted as $h_{ij} \in \mathbb{R}^N$, is obtained by:

$$\{h_{i1}, h_{i2}, ...h_{iL}\} = \text{biLSTM}_\phi(x_{i1}, x_{i2}, ...x_{iL}) \tag{1}$$

where h_{ij} is the hidden representation of input word x_{ij} learned by the encoder biLSTM_ϕ with learnable parameters ϕ. Based on the learned word representation, we aggregate the context representation of sentence i, denoted as $\boldsymbol{s}_i$, by a two-layer Multi-Layer Perceptron (MLP) based attention:

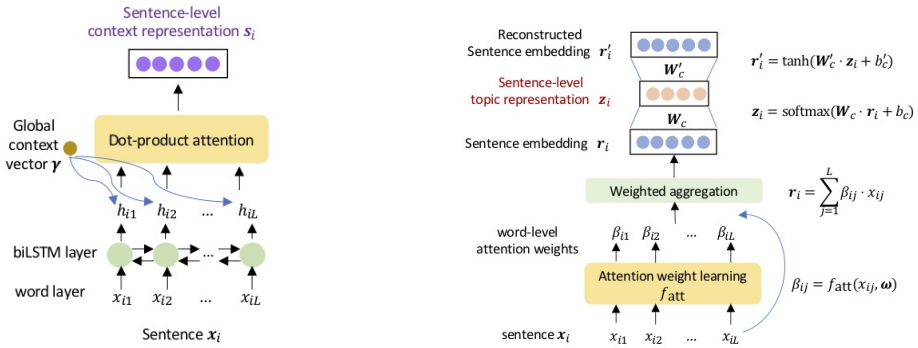
$$u_{ij} = \tanh(W_s \cdot h_{ij} + b_s), \quad \alpha_{ij} = \frac{\exp(\gamma^\intercal \cdot u_{ij})}{\sum_{j'} \exp(\gamma^\intercal \cdot u_{ij'})}, \quad \boldsymbol{s}_i = \sum_{j=1}^{L} \alpha_{ij} \cdot h_{ij} \tag{2}$$

Table 1
Notation used in the article.

Symbol	Description
Sentence Representation Learning Module	
N	The dimension of input word embeddings and learned sentence embeddings.
$x_{ij} \in \mathbb{R}^N$	The input embedding of j-th word in i-th sentence.
$\boldsymbol{s}_i \in \mathbb{R}^N$	The learned embedding of i-th sentence.
biLSTM_ϕ	The bidirectional LSTM encoder with learnable parameter set ϕ.
$u_{ij} \in \mathbb{R}^{N/2}$	The attention vector for j-th word in i-th sentence.
$a_{ij} \in \mathbb{R}$	The attention signal for j-th word in i-th sentence.
Topic Representation Learning Module	
K	The dimension of topic embeddings.
β_{ij}	The topic based attention signal for j-th word in i-th sentence.
$\boldsymbol{\omega}$	The parameter set in topic based attention learning, which includes $W_\mu, W_\omega \in \mathbb{R}^{N\times K}$, and $b_\mu, b_\omega \in \mathbb{R}^K$.
$\boldsymbol{r}_i \in \mathbb{R}^N$	The sentence representation of the i-th sentence derived based on the word-level topic attentions β_{ij}.
$\boldsymbol{z}_i \in \mathbb{R}^K$	The topic representation of the i-th sentence.
$r'_i \in \mathbb{R}^N$	The reconstructed sentence representation of the i sentence based on the learned autoencoder.
$\mathcal{R}_i, \lambda_i$	The regularization term and its corresponding weight.
z_{ik}	The probability that a sentence i belongs to the k-th topic, also represented as $P(t_k\|\boldsymbol{s}_i)$.
g_k^d	The occurrence probability of the k-th topic in document d, also defined as $P(t_k\|d)$.
Document Representation Learning Module	
c_{ij}	The similarity between the i-th and j-th sentence based on the learned topic representation.
e_{ij}	The static edge weight derived by normalizing c_{ij}.
$\boldsymbol{s}_i^l$	The representation of the i-th sentence learned by the graph attention network in the l-th iteration, where $\boldsymbol{s}_i^0$ is initialized by $\boldsymbol{s}_i$.
$\boldsymbol{w}_d$	The representation of document d.
η_a, η_b	The weight of different terms in the loss function for document representation learning.

where $W_s \in \mathbb{R}^{N\times N/2}, b_s \in \mathbb{R}^{N/2}$ are learnable parameters for the first layer MLP with the activation function `tanh`. The output of the first layer MLP, $u_{ij} \in \mathbb{R}^{N/2}$, is the attention vector of the j-th word in the i-th sentence. In the second layer MLP, we use an inner product based mapping function with `softmax` normalization to capture the attention signal of α_{ij} for u_{ij}. Here, γ is a learnable vector in the second-layer MLP and is shared among all sentences, which can be considered as a center point for label relevant representation in the latent space, or a global context vector. The similarity between u_{ij} (i.e., the word representation after the first layer MLP) and γ reflects the importance of the corresponding word in the classification. We use $\boldsymbol{s}_i$ to denote the learned context representation for the i-th sentence in a document d.

The aforementioned approach in producing the sentence-level contextual representations is a typical way in encoding sentence semantics. When used in building neural classifiers, we would expect such representations to implicitly capture the class label information. More concretely, its word-level attention weights can be used to identify words that are important for text classification decisions. Taking sentiment classification



(a) Context representation learning. (b) Topic representation learning.

Figure 3
Sentence-level context and topic representation learning.

as an example, as has been previously shown in hierarchically stacked LSTM or GRU networks, words with higher attention weights are often indicative of polarities (Yang et al. 2016) .

3.1.2 Topic Representation Learning. Sentence-level contextual representations learned in Section 3.1.1 implicitly inject label information to $\boldsymbol{s}_i$ and capture label-dependent word features. Here, we propose using a label-independent approach to capture the hidden relationships between input words to infer latent topics, which could be subsequently used to determine their potential contributions to class labels, in order to enhance the generalization and interpretability.

We propose using a Bayesian inference based autoencoder to learn the sentence-level topic representation, as shown in Figure 3b. More concretely, we assume the conditional probability of k-th topic t_k given a sentence $\boldsymbol{x}_i$, denoted as $P(t_k|\boldsymbol{x}_i)$, is obtained by the conditional probability of the corresponding topic given its constituent words x_{ij} by:

$$P(t_k|\boldsymbol{x}_i) = \sum_{j=1}^{L} P(t_k|x_{ij}) \cdot P(x_{ij}|\boldsymbol{x}_i) \tag{3}$$

In this equation, $P(t_k|x_{ij})$ can be learned by an autoencoder and $P(x_{ij}|\boldsymbol{x}_i)$ is obtained by a Bayesian inference approach. In particular, we denote $P(x_{ij}|\boldsymbol{x}_i)$ as β_{ij}, which can be considered as the topic-based attention weight of a word x_{ij} in a sentence $\boldsymbol{x}_i$. It is a latent variable and its value is given by a stochastic generative process:

$$\beta_{ij} = f_{\text{att}}(x_{ij}, \boldsymbol{\omega}), \quad \boldsymbol{\omega} \sim \mathcal{N}(\mu_\omega, \sigma^2_\omega) \tag{4}$$

where x_{ij} is the embedding of the j-th word in i-th sentence, and $\boldsymbol{\omega}$ denotes the parameters for attention weight learning. More generally, we aim to learn $p(\boldsymbol{\omega}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\omega})p(\boldsymbol{\omega})$, where $\mathcal{D}$ denotes all the training documents, and $\boldsymbol{\omega}$ is sampled from the

variational posterior $q(\boldsymbol{\omega}|\mathcal{D})$, which also assumes following a Gaussian distribution and can be approximated by a neural network, namely,

$$\boldsymbol{\mu}_\omega = f_\mu(x_{ij}) = \text{sigmoid}(W_\mu \cdot x_{ij} + b_\mu) \tag{5}$$

$$\boldsymbol{\sigma}^2_\omega = f_\sigma(x_{ij}) = \text{sigmoid}(W_\sigma \cdot x_{ij} + b_\sigma) \tag{6}$$

$$\epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad f_{\text{att}}(x_{ij}, \boldsymbol{\omega}) = \boldsymbol{\mu}_\omega + \boldsymbol{\sigma}_\omega \cdot \epsilon \tag{7}$$

Here, f_μ, f_σ are MLP-based approximation to obtain the mean and variance of the input word representation, $W_\mu, W_\omega \in \mathbb{R}^{N\times K}$ and $b_\mu, b_\omega \in \mathbb{R}^K$ are learnable parameters in the MLP layer, and $\mathbf{I}$ is the identity matrix. The parameters in these layers are shared across all input words.

Once the word-level attention weights are learned, the sentence embedding, denoted as $\boldsymbol{r}_i \in \mathbb{R}^N$, is obtained by $\boldsymbol{r}_i = \sum_{j=1}^{L} \beta_{ij} \cdot x_{ij}$. We then feed the sentence-level representation $\boldsymbol{r}_i$ into an autoencoder to generate the reconstructed representation $\boldsymbol{r}'_i$:

$$\boldsymbol{z}_i = \text{softmax}(\boldsymbol{W}_c \cdot \boldsymbol{r_i} + \boldsymbol{b}_c), \quad \boldsymbol{r}'_i = \tanh(\boldsymbol{W}'_c \cdot \boldsymbol{z_i} + \boldsymbol{b}'_c) \tag{8}$$

where $\boldsymbol{W}_c, \boldsymbol{W}'_c \in \mathbb{R}^{N\times K}, \boldsymbol{b}_c, \boldsymbol{b}'_c \in \mathbb{R}^K$ are learnable parameters that can be used to generate topics, $\boldsymbol{z}_i \in \mathbb{R}^K$ is the hidden topic vector that is considered as the sentence-level topic distribution for i-th sentence, and $\boldsymbol{r}'_i \in \mathbb{R}^N$ is the reconstructed sentence representation for i-th sentence based on the corresponding topic representation $\boldsymbol{z}_i$. Because $p(\mathcal{D}|\boldsymbol{\omega})$ is intractable, we resort to neural variational inference to maximize the Evidence Lower BOund (ELBO) that

$$\mathcal{L}_e(\boldsymbol{w}_d) = \sum_{i=1}^{M_d} \log p(\boldsymbol{r}'_i|x_{ij}, \boldsymbol{\omega}) - D_{KL}(q(\boldsymbol{\omega}|\mathcal{D})||p(\boldsymbol{\omega})) \tag{9}$$

where $\boldsymbol{w}_d$ denotes the representation for document d. The first term denotes the reconstructed sentence representation, and the second term is the KL diversity measuring the difference between the variational posterior and the prior distribution. The posterior distribution is learned from the training corpus $\mathcal{D}$, while the prior distribution of $\boldsymbol{\omega}$ is a normal distribution.[2]

Regularization Terms. In the following, we introduce a number of regularization terms used in our model.

Orthogonal regularization. To make learned sentence-level topic representation different from the sentence-level context representation, we simultaneously minimize the inner product between the reconstructed sentence representation $\boldsymbol{r}'_i$ and the context representation $\boldsymbol{s}_i$. Hence, we define an orthogonal regularization term below:

$$\mathcal{R}_1(\boldsymbol{w}_d) = \sum_{i}^{M_d} ||\boldsymbol{r}'_i \cdot \boldsymbol{s}_i||_2 \tag{10}$$

2 Note that $\boldsymbol{\omega}$ is shared among all inputs, which is different from typical latent variable models in which a local latent variable is associated with each individual input.

Topic uniqueness regularization. Note that the learned topics might be redundant, that is, different topics might contain many overlapping words. To ensure the diversity of the resulting topics learned, we add a regularization term to the objective function to encourage the uniqueness of each topic embedding.

$$\mathcal{R}_2(\boldsymbol{w}_d) = ||\boldsymbol{W'}_c \cdot \boldsymbol{W'}_c^{\intercal} - \mathbf{I}||_2 \tag{11}$$

where $\mathbf{I}$ is the identity matrix, and $\boldsymbol{W'}_c$ is the decoder matrix defined in Equation (8), where each column can be extracted as the representation of a topic. $\mathcal{R}_2(\boldsymbol{w}_d)$ reaches its minimum value when the dot product between any two different topic representations is zero.

Topic discrepancy regularization. Based on the topic representations, $\boldsymbol{z}_i$, we can essentially partition text into different groups. Inspired by Johansson, Shalit, and Sontag (2016), we propose another regularization term to re-weigh different partitions.

Intuitively, we want to reduce the discrepancy between different latent topics weighted by the posterior probability of topics' given text in order to prevent the learner from using "unreliable" topics of the data when trying to generalize from the factual to the counterfactual domains. For example, if in our movie reviews, very few people mentioned the topic of "*source effect*," inferring the attitude toward this topic is highly prone to error. As such, the importance of this topic should be down-weighted.

Without loss of generality, assuming an input sentence $\boldsymbol{x}_i$ contains L words, $\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{iL}\}$, where x_{ij} is the word embedding of the j-th word in i-th sentence. Each sentence is mapped to a latent topic distribution with K dimensions, $\boldsymbol{z}_i = \{z_{i1}, z_{i2}, \ldots, z_{iK}\}$, with each of its elements representing the probability that the input sentence i belongs to the k-th topic, $z_{ik} = P(t_k|\boldsymbol{x}_i)$. The occurrence probability of the k-th topic in document d is defined as:

$$P(t_k|d) = \frac{1}{M_d}\sum_{i=1}^{M_d} P(t_k|\boldsymbol{x}_i) = \frac{1}{M_d}\sum_{i=1}^{M_d} z_{ik} = g_k^d \tag{12}$$

Inspired by Johansson, Shalit, and Sontag (2016), in which the discrepancy is defined as a function of the distance between the weighted population means, we define the discrepancy between two topics, t_a, t_b, in a document d as the distance between two topic representations weighted by their occurrence probabilities in document d:

$$\begin{aligned} disc(t_a, t_b) &= 1 - \cos\big(P(t_a|d)\zeta_{t_a}, P(t_b|d)\zeta_{t_b}\big) \\ &\propto 1 - (g_a^d \cdot W_{ca})(g_b^d \cdot W'_{cb})^{\intercal} \end{aligned} \tag{13}$$

where $\cos(\cdot)$ denotes the cosine similarity function and ζ_{t_a} denotes the representation of the topic t_a (similarly for ζ_{t_b}), which is equivalent to W_{ca}, the a-th column of the encoder matrix. A brief explanation of why W_{ca} can be considered as the representation of topic t_a will be given in Section 4. In Equation (13), the topic representations W_{ca} and W_{cb} are global and are shared across all documents, while the topic occurrence probabilities g_a^d and g_b^d are local and are specific to document d. To understand the effect of applying a regularization term defined in Equation (13), we illustrate below the derivation of the gradient on topic t_a's representation ζ_{t_a}. First, assuming the regularization term defined based on Equation (13) is $\mathcal{R}(disc(t_a, t_b))$, then the corresponding gradient on topic t_a's

representation ζ_{t_a} is $\frac{\partial \mathcal{R}\left(disc(t_a,t_b)\right)}{\partial disc(t_a,t_b)} \cdot \frac{\partial disc(t_a,t_b)}{\partial \zeta_{t_a}}$. Without loss of generality, we do not give a specific formula of the regularization function here and only focus on the second component, $\frac{\partial disc(t_a,t_b)}{\partial \zeta_{t_a}}$:

$$\begin{aligned}
\frac{\partial disc(t_a,t_b)}{\partial \zeta_{t_a}} &= -\Big(\frac{\partial g_a^d \cdot W_{ca}}{\partial W_{ca}}\Big) \cdot (g_b^d \cdot W_{cb})^\intercal \\
&= -\Big(\frac{\partial \sum_{i=1}^{M_d} \boldsymbol{s}_i \cdot W_{ca}^\intercal \cdot W_{ca}}{\partial W_{ca}}\Big) \cdot \frac{(g_b^d \cdot W_{cb})^\intercal}{M_d} \\
&= -2 \cdot g_b^d \frac{\sum_{i=1}^{M_d} \boldsymbol{s}_i \cdot W_{ca} \cdot W_{cb}^\intercal}{M_d} \\
&\propto -2 \cdot g_b^d \cdot \cos(\zeta_{t_a}, \zeta_{t_b}) \frac{\sum_{i=1}^{M_d} \boldsymbol{s}_i}{M_d}
\end{aligned} \tag{14}$$

The above result states that if two input topics have similar representations, or have higher occurrence probabilities in the current document, they will obtain larger updates that push their representations closer to the representation of the current document calculated based on the mean pooling of its constituent sentences. Essentially, the discrepancy term separates the document representations based on the topic similarities, then updates the corresponding topic distribution based on the topic occurrence probabilities and the document representation.

We can extend the discrepancy term to cater to all possible topic pairs as:

$$Disc_d = \begin{bmatrix} disc(t_1,t_1) & disc(t_1,t_2) & \cdots & disc(t_1,t_K) \\ disc(t_2,t_1) & disc(t_2,t_2) & \cdots & disc(t_2,t_K) \\ \vdots & \vdots & \ddots & \vdots \\ disc(t_K,t_1) & disc(t_K,t_2) & \cdots & disc(t_K,t_K) \end{bmatrix} = \mathbf{1}_{K\times K} - P_m(d) \odot \boldsymbol{W}_c' \boldsymbol{W}_c'^\intercal \tag{15}$$

where $\mathbf{1}_{K\times K}$ is a $k \times k$ matrix in which all elements are 1, $\odot$ is the element-wise product, and $P_m(d)$ is defined as:

$$P_m(d) = \begin{bmatrix} g_1^d \cdot g_1^d & g_1^d \cdot g_2^d & \cdots & g_1^d \cdot g_K^d \\ g_2^d \cdot g_1^d & g_2^d \cdot g_2^d & \cdots & g_2^d \cdot g_K^d \\ \vdots & \vdots & \ddots & \vdots \\ g_K^d \cdot g_1^d & g_K^d \cdot g_2^d & \cdots & g_K^d \cdot g_K^d \end{bmatrix} \tag{16}$$

We can then define the discrepancy term based regularization by l_2 norm as $\|\mathbf{1}_{K\times K} - P_m(d) \odot \boldsymbol{W}_c' \boldsymbol{W}_c'^\intercal\|_2$. The gradient of such a regularization term with respect to a specific topic representation would guide its movement toward the centroid of the other topics weighted by their occurrence probabilities defined in Equation (12). While the topic uniqueness regularization term defined in Equation (11) aims to ensure the orthogonality among topics, the discrepancy term based regularization will push the representations of major topics closer to the input document representation.

In practice, to achieve the balance between two regularization terms, we combine the discrepancy term based regularization with the topic uniqueness regularization defined in Equation (11) as:

$$\mathcal{R}_2(\boldsymbol{w}_d) = \|\boldsymbol{W}'_c \cdot \boldsymbol{W}'^{\intercal}_c - \big(\alpha \cdot \mathbf{I} + (1-\alpha) \cdot P_m^{-1}(d)\big)\|_2 \tag{17}$$

where $\alpha \in (0,1)$ determines the contribution of the topic uniqueness term. For any $P_m^{-1}(d)_{ij}$, which denotes the the i-th row and j-th column element in $P_m^{-1}(d)$, $P_m^{-1}(d)_{ij} = 1/(g_i^d \cdot g_j^d)$. In our experiments, we set $\alpha = 0.5$.

Final objective function. The final objective function L_{topic} for sentence-level topic representation learning is defined below, where λ_1 and λ_2 are used to control the relative contributions of different terms.

$$\mathcal{L}_{\text{topic}}(\boldsymbol{w}_d) = \mathcal{L}_e(\boldsymbol{w}_d) + \lambda_1 \cdot \mathcal{R}_1(\boldsymbol{w}_d) + \lambda_2 \cdot \mathcal{R}_2(\boldsymbol{w}_d) \tag{18}$$

3.2 Document Modeling

After obtaining the sentence-level context representations $\boldsymbol{s}_i$ and latent topic representations $\boldsymbol{z}_i$, the next step is to aggregate such representations to derive the document-level representation for classification.

Graph Node Update. We represent each document by a graph in which the nodes represent sentences $\{\boldsymbol{s}_i\}_{i=1}^{M_d}$ in the document d and the edges linking every two nodes measure their topic similarities. Each graph node is initialized by its respective sentence-level context representation $\boldsymbol{s}_i$. The topic similarity c_{ij} between the i-th sentence and j-th sentence is defined as the inner-product of their latent topic vectors, $c_{ij} = \boldsymbol{z}_i^{\intercal}\boldsymbol{z}_j$.

Graph Attention Networks (GATs) are applied to update the graph nodes. Classic GATs learn attention weights via applying self-attention on node features, and update the edge weights during training. Here, we leverage the normalized topic similarity $e_{ij} = \texttt{softmax}(c_{ij})$ as the static edge weight and aggregate the sentences sharing the similar topics. In this way, sentences linked with larger weight edges are topically more similar. The graph node features are updated as follow:

$$\boldsymbol{s}_i^{\ell+1} = \texttt{Relu}\left(\sum_{j \in \mathcal{N}_i} e_{ij} \boldsymbol{W} \boldsymbol{s}_j^{\ell}\right) \tag{19}$$

where $\boldsymbol{s}_i^{\ell+1}$ denotes the hidden representation of node (or sentence) $\boldsymbol{x}_i$ in the $(\ell+1)$th iteration and $\boldsymbol{s}_i^0$ is initialized by the context sentence representation $\boldsymbol{s}_i$ learned from Context Representation Learning module, $\mathcal{N}_i$ denotes the neighbors of node i, and $\boldsymbol{W}$ is the learnable weight matrix for the graph nodes.

Document Classification. For a document d given the last layer output from its graph, $\{\boldsymbol{s}_i^L\}_{i=1}^{M_d}$, we average the M_d node representations as the document representation, $\boldsymbol{w}_d = (\boldsymbol{s}_1^L + \boldsymbol{s}_2^L ... + \boldsymbol{s}_{M_d}^L)/M_d$. The document representation is fed to our classification layer

(i.e., softmax) to generate the predicted outputs, $\hat{y} = \texttt{softmax}(\boldsymbol{w}_d)$. The classification loss is defined as:

$$\mathcal{L}_c(\boldsymbol{w}_d) = -\sum_{c=1}^{C} y_c \cdot \log(\hat{y}_c) \tag{20}$$

where C denotes the total number of class categories. The final loss function is defined as:

$$\mathcal{L}_{final}(\boldsymbol{w}_d) = \eta_a \cdot \mathcal{L}_{\text{topic}}(\boldsymbol{w}_d) + \eta_b \cdot \mathcal{L}_c(\boldsymbol{w}_d) \tag{21}$$

where η_a and η_b are the weights to control the contribution of the respective loss to the final objective function.

4. Model Interpretation Generation

For a given document d, the proposed model can not only predict a label, but also generate a hierarchical interpretation for its prediction. Taking the document in Figure 1 as an example, we will elaborate below how to generate the word- and sentence-level explanations, as well as how to aggregate the hierarchical information to generate the final model prediction.

Word-Level Interpretation Generation. The dual module learns a context representation and a topic representation of a sentence. The approach in producing the context representations is a typical way in encoding sentence semantics (§3.1.1). When used in building neural classifiers, we would expect that such representations implicitly capture the class label information. More concretely, its word-level attention weights can be used to identify words that are associated with the class label. For the example shown in Figure 1, label-relevant words such as "*worst*" is indicative of the *negative* polarity. The latent topic learning module (§3.1.2) aims to capture latent topics shared across all documents regardless of their class labels. The word-level attention weights are generated by a stochastic process as shown in Equation (4). It can be observed that words identified in this way are more topic-related (such as "*production*" and "*actors*") and are less relevant to the class label.

To extract topic words (the word cloud in Figure 1), we first multiply the word embeddings matrix $E \in \mathbb{R}^{V \times N}$ with the weight matrix $\boldsymbol{W}_c \in \mathbb{R}^{N \times K}$ from the topic encoder network (Equation (8) in §3.1.2), where V denotes the vocabulary size, N is the word embedding dimension, and K is the topic number.[3] From the resulting matrix $\boldsymbol{\pi} \in \mathbb{R}^{V \times K}$, we can then extract the top n words from each topic dimension as the topic words (Chaney and Blei 2021). In the following, we explain why each column in $\boldsymbol{\pi}$ can be considered as a topic.

In Section 3.1, we assume the topic distribution is obtained by an encoder-decoder formulation in Equation (8), its topic representation for sentence $\boldsymbol{x}_i$, denoted as $\boldsymbol{z}_i$, is a K-dimension vector of $\boldsymbol{z}_i = \{z_{i1}, z_{i2}, \ldots, z_{iK}\}$, with each of its elements representing the

3 We can also use the decoder weight matrix $\boldsymbol{W}'_c$, which is symmetrical to $\boldsymbol{W}_c$.

probability that the input sentence belongs to the k-th topic, $z_k = P(t_k|\boldsymbol{x})$, and the encoder layer can be rewritten as a softmax function which generates a probability:

$$z_{ik} = \frac{\exp\left(W_{ck}^{\intercal} \cdot \sum_j x_{ij} \cdot P(x_{ij}|\boldsymbol{x}_i) + b_{ck}\right)}{\sum_m \exp\left(W_{cm}^{\intercal} \cdot \sum_j w_j \cdot P(x_{ij}|\boldsymbol{x}_i) + b_{cm}\right)}$$

where W_{ck} is the k-th column of the encoder matrix $\boldsymbol{W}_c = \{W_{c1}, W_{c2}, \ldots, W_{cK}\}$, and $\boldsymbol{b}_c = \{b_{c1}, b_{c2}, \ldots, b_{cK}\}$ is the bias term. We then have:

$$z_k \propto W_{ck}^{\intercal} \cdot \sum_j x_{ij} \cdot P(x_{ij}|\boldsymbol{x}_i) + b_{ck} \propto W_{ck}^{\intercal} \cdot \sum_j x_{ij} \cdot P(x_{ij}|\boldsymbol{x}_i) \tag{22}$$

Because the activation function in our network is bijection, we can simply take $z_k \propto W_{ck}^{\intercal} \cdot \boldsymbol{x}_i$ to guarantee that Equation (22) is correct. Therefore, we can use the corresponding column in the encoder matrix to search the whole vocabulary to identify the top-n words in each topic.

Sentence-Level Interpretation Generation. From the sentence-level latent topic representation, we can identify the most prominent topic dimension in the hidden topic vector z_i and use it as the topic label for each sentence. As has been illustrated in the lower part (word cloud) in Figure 1, the topics that correspond to the six sentences can be summarized as "*Awful movie*," "*Actor performance*," "*Injury*," and "*Inconsistent Plot*," from left to right. Here, we represent each topic as a word cloud, which contains the top-10 topic-associated words from the corpus vocabulary as shown in Figure 1. The topic labels are manually assigned for better illustration. We can also automatically generate topic labels by selecting the most relevant phrase from the document to represent each topic (see in Figure 10). Specifically, for each topic, we first find the most relevant sentence according to the sentence-topic distribution, that is, $\mathcal{T}^{M\times K} = S^{M\times d} \times (Z^{K\times d})^{\intercal}$, where M is the number of sentences in a document, d is the dimension of a sentence representation, and S and Z are the sentence contextual representations and the topic representations, respectively. Then, we extract the key phrase from the sentence via the Rapid Automatic Keyword Extraction algorithm.[4] We infer the class label of each sentence by feeding the sentence contextual representation into the classification layer of HINT and obtain probabilities of class labels, thus obtaining its class-associated intensity.

Document-Level Interpretation Generation. Once the topic and class label for each sentence is obtained, we can aggregate sentences based on the similarity of their latent topic representations. The contextual representation of the document is obtained by taking the weighted aggregation of its constituent sentence contextual representations, where the weights are the topic similarity values. As sentences are assigned to various topics, we can easily study how topics and their associated class labels change throughout the document. In addition, we can also infer the most prominent topic in the document.

4 https://pypi.org/project/rake-nltk/.

Table 2
Dataset statistics. Guardian News has the largest average document length and the most imbalanced class distribution.

Datasets	Avg. Length	Class ratio	#Train	#Test
Yelp	139	2:1	140k	20k
IMDB	218	1:1	25k	25k
Guardian	1,024	4:2.5:2:1:0.5	37.03k	15.87k

5. Experimental Setup

5.1 Datasets

We conduct experiments on three English document datasets, including two review datasets: patient reviews extracted from Yelp[5] and the IMDB movie reviews (Maas et al. 2011), as well as the Guardian News dataset.[6] For Yelp reviews, we retrieve patient reviews based on a set of predefined keywords.[7] Each review is accompanied by keywords indicating its associated healthcare categories. Because the majority of reviews have ratings of either 1 or 5 stars, we only keep the reviews with 1 and 5 stars as negative and positive instances, respectively. The IMDB dataset also has two class categories (*positive* and *negative*). Yelp has twice as many positive reviews as negative ones while IMDB has a balanced class distribution. As the IMDB dataset does not provide the train/test split, we follow the same split proportion as that in the implementation of Scholar.[8] The Guardian News dataset contains 5 categories, namely, *Sports*, *Politics*, *Business*, *Technology*, and *Culture*, from which nearly 40% of the documents are in the *Sports* category and less than 10% and 5% of the documents are in the *Technology* and *Culture* categories, respectively. The data statistics are shown in Table 2.

5.2 Baselines

We compare our approach with the following baselines:

- *CNN*: In our experiments, the kernel sizes are 3,4,5, and the number of kernels of each size is 100.
- *LSTM*: For each document, words are fed into LSTM sequentially and composed by mean pooling. A softmax layer is stacked to generate the class prediction. We also report the results of LSTM with an attention mechanism (LSTM+Att).
- *HAN* (Yang et al. 2016): The Hierarchical Attention Network stacks two bidirectional Gated Recurrent Units and applies two levels of attention mechanisms at the sentence-level and at the document-level, respectively.

5 https://www.yelp.com/dataset.
6 https://www.kaggle.com/sameedhayat/guardian-news-dataset/tasks.
7 All the keywords are listed in Appendix B.
8 https://github.com/dallascard/scholar.

- *BERT* (Devlin et al. 2019): We feed each document into BERT as a long sequence with sentences separated by the `[SEP]` token, which is fine-tuned on our data. We truncate documents with length over 512 tokens and use the representation of the `[CLS]` token for classification.
- *Scholar* (Card, Tan, and Smith 2018): A neural topic model trained with variational autoencoder (Kingma and Welling 2014) with document-level class labels incorporated as supervised information. Scholar essentially learns a latent topic representation of an input document and then predicts the class label conditional on the latent topic representation.
- *VMASK* (Chen and Ji 2020): The model applies variational word masking strategy to mask out unimportant words to improve interpretability of neural text classifiers. During training, the binary mask is derived from the *Gumbel-softmax* operator on the non-linear transformation of an input sentence, and then element-multiplication is applied on the mask and the sentence to remove the unimportant words. In inference, they use softmax to get a softened version of the mask, instead. We report results from two variants of VMASK, by using the text input encoded either by BERT or LSTM.[9]

5.3 Data Pre-processing

For the IMDB reviews, we use the processed IMDB dataset provided by Scholar.[10] For both review datasets, we set the maximum sentence length to 60 words and the maximum document length to 10 sentences. We only keep the most frequent 15,000 words in the training set, and mark the other words as [unk]. Sentences with more than 30% [unk] are removed from our training set. For the Guardian news data, we download the dataset from Kaggle[11] and follow its provided train/test split. We set the maximum sentence length to 60 words and the maximum document length to 18 sentences.

6. Experimental Results

6.1 Text Classification Results

The text classification results are shown in Table 3. Methods marked with † are re-implemented by us. As shown in Table 3, the vanilla classification models, such as CNN and LSTM, show inferior performance across three datasets. With the incoporation of the attention mechanism, LSTM-att slightly improves over LSTM. HAN was built on bidirectional GRUs but with two levels of attention mechanism at the word- and the sentence-level. It outperforms LSTM-att. BERT was built on the Transformer architecture. But it gives slightly worse results compared to HAN. The hierarchical modeling in HAN may explain its comparatively superior performance. The neural topic modeling approach, Scholar, performs better than CNN, but slightly worse than other baselines on IMDB and Yelp. VMASK learns to assign different weights to word-level features

9 Our results on IMDB are different from those reported in the original paper due to different train/test splits.
10 IMDB dataset download script.
11 Guardian News dataset.

Table 3
Classification accuracy on the three datasets. ** significant at $p < 0.05$, *** significant at $p < 0.001$.

Methods	IMDB	Yelp	Guardian
CNN†	83.36	94.16	92.82
LSTM†	87.30	97.10	93.57
LSTM-att†	87.56	97.30	93.97
HAN	87.92	97.70	94.34
BERT	87.59	97.52	94.28
Scholar	86.10	96.87	93.97
VMASK-BERT	88.23***	98.10**	94.49**
VMASK-LSTM	87.40	98.04	93.79
HINT	**89.11****	**98.42****	**95.38****

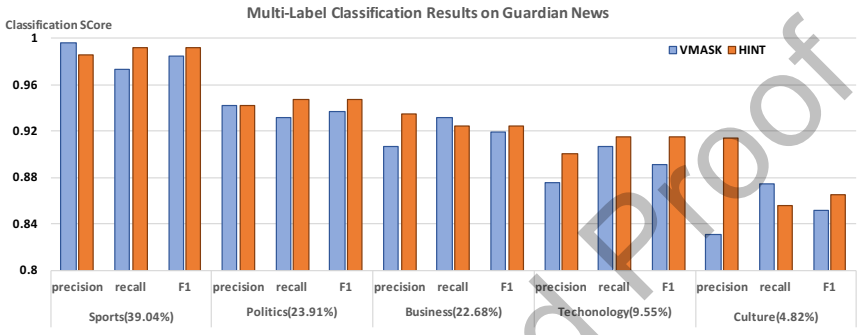


Figure 4
The precision, recall, and F1 of per-class classification results of VMASK and HINT on the Guardian News dataset.

by minimizing the classification loss. Its BERT variant generally outperforms the LSTM variant and gives the best results among the baselines. We have additionally performed a statistical significance test, the Student's *t*-test, to compare the performance of HINT with VMASK-BERT by training both models for 10 times, and show the results in Table 3. In general, HINT outperforms all baselines and the improvement is more prominent on the largest Guardian News dataset with the longest average document length.

To further examine the ability of HINT in dealing with imbalanced data, we plot in Figure 4 the per-class precision, recall, and F1 results on the Guardian News data. While HINT generally outperforms VMASK in F1 across all classes, it achieves much better results on minority classes. For example, HINT improves upon VMASK by nearly 8% in precision on the smallest *Culture* class.

6.2 Topic Evaluation Results

The sentence-level topic representation learning module in the HINT framework generates latent topic vectors that allow us to extract top associated words for each latent topic dimension by the weights of the decoder layer which transforms the latent variables to the reconstructed input. Existing work shows that good latent variables should be able to cluster the high-dimensional text representations into coherent semantic groups

(a) Example topic words from Yelp.

(b) Example topic words from IMDB.

Figure 5
The topic word importance is indicated by the word font size. In each sub-figure, the upper topic word clouds are positive, while the lower topic word clouds are negative.

(Kingma and Welling 2014). As described in Section 4, we can interpret the top associated words for each latent topic dimension as topic words. In this subsection, we show the topic extraction results by displaying the top 10 words in each latent dimension as a word cloud.

Figures 5a and 5b show the word clouds of the generated example topic words on the Yelp and the IMDB, respectively. It can be easily inferred from Figure 5a that users express general positive comments, praising convenient facility locations and competitive pricing; while they complain about dusty environment and service quality and express negative feeling relating to their diseases. In Figure 5b, we can observe reviewers' attitudes toward different genres of movies. They like *thriller* and *animated movie*. On the contrary, they show negative feelings toward luxury *lifestyle* or movies relating to *misogyny*. These results show that HINT can indeed extract topics discussed under different polarity categories despite using no topic-level polarity annotations for topic learning. Figure 6 shows example topic words, each of which corresponds to the five news categories from the Guardian news data.

In addition to visualizing the extracted topics, we also evaluate the quality of the extracted topics using four different topic coherence measures, including the normalized Pointwise Mutual Information (NPMI), a lexicon-based method (UCI), and context-vector-based coherence measures (CV). We compare the results with LDA (Blei, Ng, and Jordan 2003) and Scholar (Card, Tan, and Smith 2018) in Table 4. It can be observed

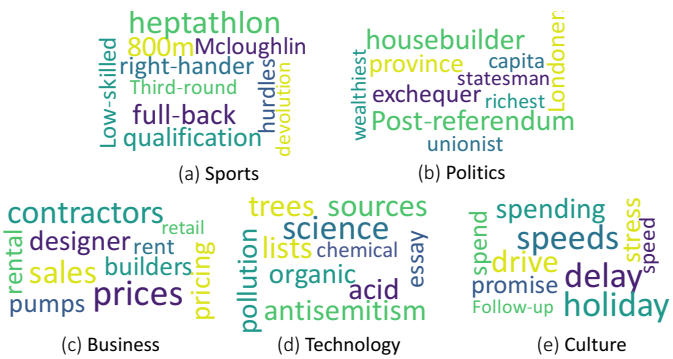


Figure 6
Example topic words for each of the news categories in the Guardian News dataset.

Table 4
Topic coherence results.

Method	IMDB			Yelp			Guardian News		
	CV	NPMI	UCI	CV	NPMI	UCI	CV	NPMI	UCI
LDA	0.341	−**0.032**	−**1.936**	0.377	−**0.039**	−1.495	0.362	−0.140	−2.858
Scholar	0.351	−0.057	−2.010	0.424	−0.061	−2.188	0.373	−0.286	−**1.207**
HINT	**0.401**	−0.068	−1.992	**0.445**	−**0.039**	−**1.385**	**0.423**	−**0.108**	−3.085

that overall, HINT gives the best results on Yelp. It performs worse than Scholar on the Guardian News in UCI, but achieves better results in CV and NPMI. On the IMDB dataset, however, HINT only outperforms the other two models in CV and was beaten by LDA in both NPMI and UCI. One possible reason is that HINT estimates the word probability by context embedding. Hence, it beats the baselines on context-vector-based coherence, but only achieves comparable performance on the lexicon-based metric.

6.3 Interpretability Evaluation

A good interpretation method should give explanations that are (i) easily understood by humans and (ii) indicative of *true* importance of input features. We conduct both quantitative and human evaluations on the interpretation results generated by HINT.

6.3.1 Word Removal Experiments. A good interpretation model should be able to identify truly important features when making predictions (Alvarez-Melis and Jaakkola 2018). A common evaluation strategy is to remove features identified by the interpretation model, and measure the drop in the classification accuracy (Chen and Ji 2020).

Figure 7 shows the correlation score between the accuracy drop and the number of removed words evaluated on the three datasets. In addition to VMASK, we also take two variants of HINT as the contrasts, namely, HINT-Context and HINT-Topic. The

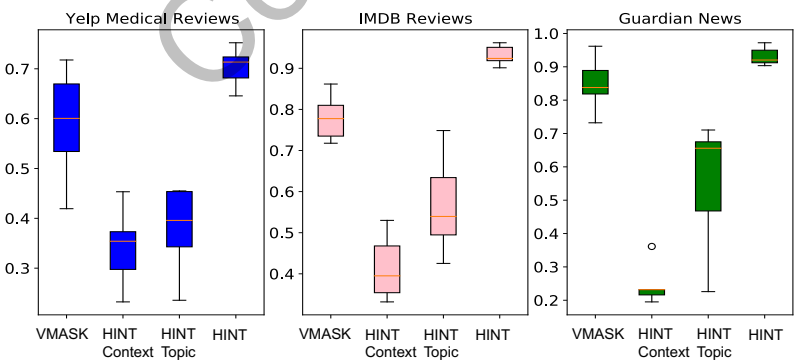


Figure 7
Aggregated correlation score between the classification accuracy drop and the number of removed words. HINT shows the highest correlation score, that is, removing the top topic words identified by HINT leads to a more significant performance drop compared to word masking methods.

Table 5
Accuracy drop by VMASK and HINT with different numbers of removed words.

k	ACC↓	Method	Removed words
20	1.1%	VMASK	brilliantly best intelligently tough cynicism
	1.2%	HINT	unwatchable highest dramas deeply flawless
40	2.8%	VMASK	interesting timeless lacks failed remaining
	3.8%	HINT	perfection comedies screenwriter reporter disagree
60	2.3%	VMASK	recommend like pretty suggestion poorly
	4.6%	HINT	scripts mysteries complaint funeral werewolf

former masks the words assigned with large α_{ij} weights by the context learning module; the latter masks the words with large β_{ij} attention weights in the topic learning module. HINT masks the top-K unique topic words according to their weights in each topic. It can be observed that simply masking words with higher weights identified by the context learning module or the topic learning module does not give good correlation scores. The results are worse than VMASK, which automatically determine which words to mask based on the information bottleneck theory. Nevertheless, when masking words based on those identified by HINT, we observe better correlation scores with smaller spreads compared to VMASK, showing the effectiveness of HINT in identifying task-important words.

In Table 5, we list part of the top k words removed by HINT and VMASK on the IMDB dataset with the corresponding performance drops. It can be observed that when k is 20, both methods tend to identify opinion words as key features for removal that are task-relevant, resulting in a similar classification accuracy drop. With the increasing number of k, VMASK still primarily focuses on opinion words, which leads to a further modest accuracy drop. On the contrary, when k increases, HINT starts to extract topic-related words such as "*comedies*" and "*screenwriter*." These words may seem to be task-irrelevant. However, the removal of them causes more noticeable accuracy drop. We speculate that such words are highly relevant with the latent topics discussed in text, which in turn are associated with implicit polarities important for the decision of document-level sentiment classification.

6.3.2 Human Evaluation for Interpretablity. We conduct human evaluation to validate the interpretablity of our proposed method on the following criteria inspired by existing methods on human evaluation (Zhou et al. 2020):

- *Correctness*. It measures to what extent users can make correct predictions given the model interpretations. That is, users are asked to predict the document label based on the model-generated interpretation. If the interpretation is correct, then users should be able to predict the document label easily.
- *Faithfulness.* It measures to what extent the generated explanation is faithful to the model prediction.
- *Informativeness*. It measures to what extent the interpretation reveals the key information conveyed in text such as the main topic discussed in text, its associated polarity, and the secondary topic (if there is any) mentioned in text.

We randomly select 100 samples with the interpretations generated by HAN (Yang et al. 2016), VMASK (Chen and Ji 2020), and our model for evaluation. We invite three evaluators, all proficient in English and with at least MSc degrees in Computer Science, to score the interpretations generated on the sampled data on a Likert scale of 1 to 5. Details of the evaluation protocol are presented in Appendix A.

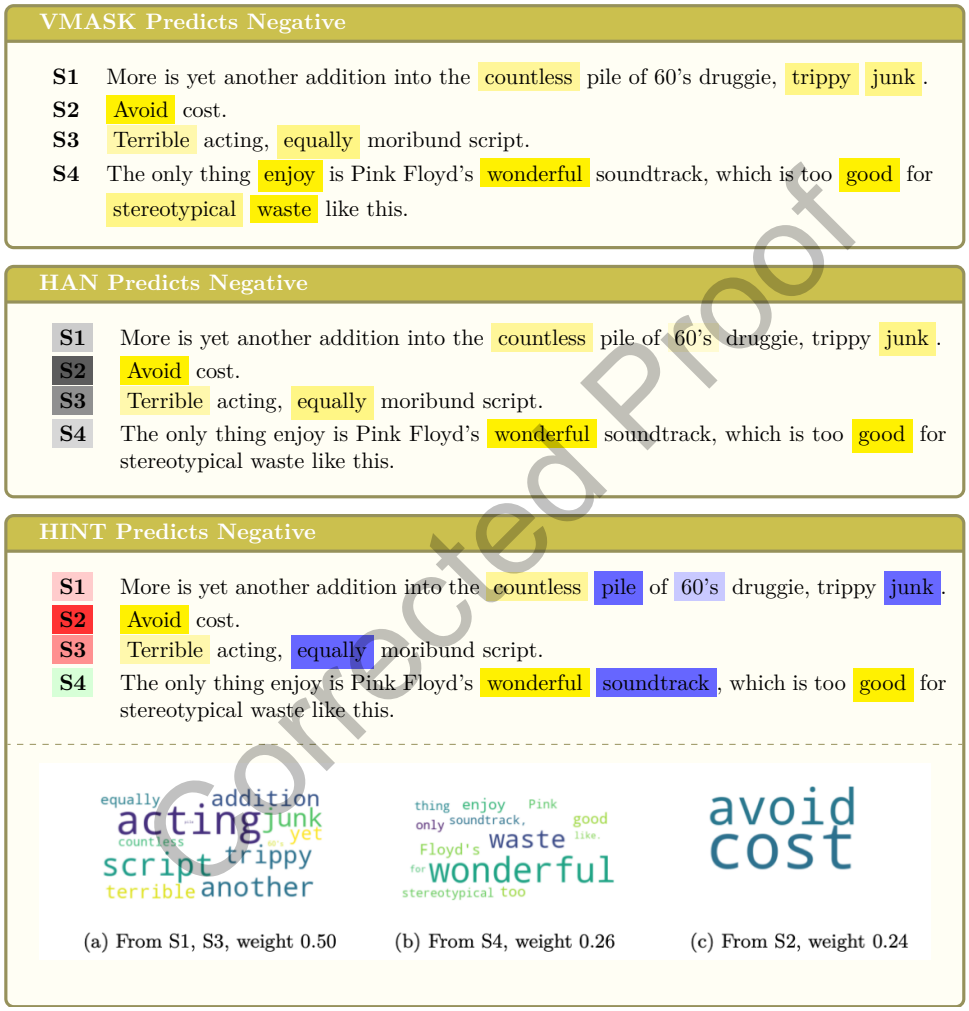


Figure 8
Interpretations generated by VMASK, HAN, and HINT on the same IMDB review with mixed sentiments. VMASK only highlights important words for classification, while HAN additionally displays sentence importance. The interpretations generated by HINT contain richer information. HINT highlights both label-independent words (in blue) and label-dependent words (in yellow) as the word-level interpretations; it also displays the sentiment strength (red for negative, green for positive) for each sentence as the sentence-level interpretations. Moreover, HINT groups sentences into three topics (shown as three word clouds) with the document-level topic weights. Along with the sentence-level sentiment, we can easily tell that the document contains mixed sentiments, and the most predominant topic (associated with S1, S3) is negative, thus inferring the document as negative.

Table 6
Human evaluation results in a likert scale of 1 to 5 (1: *Strongly Disagree*; 5: *Strongly Agree*). The inner-rater agreement measured by Kappa score is 0.37.

Model	**Correctness**	**Faithfulness**	**Informativeness**
HAN	3.89	3.92	3.79
VMASK	4.13	4.06	3.93
HINT	**4.37**	**4.28**	**4.11**

We show interpretations generated from each of the three models in Figure 8 for a movie review with mixed sentiments. The review expresses a negative polarity toward the topic of *acting* and *script*, while a positive polarity for the *soundtrack*, resulting in an overall negative sentiment. For *Correctness*, the evaluators are required to predict the document label only based on the generated interpretations without reading the document content in detail. We can observe that VMASK highlights both positive and negative words (e.g., "*junk*" and "*enjoy*"), making it relatively difficult to infer the document-level sentiment label. HAN additionally provides the sentence-level importance from which we know that sentence S2 is more important than the others and it contains the negative word "*avoid*." Compared with the baselines, HINT reveals much richer information. One can easily tell that the document contains mixed sentiments as the first three sentences carry a negative sentiment while the last one bears a positive polarity. In addition, the document discusses three topics (shown as three word clouds) with S1 and S3 associated with the most prominent topic about *acting* and *script*, carrying a negative sentiment. Thus, it seems that the HINT-generated interpretations align with the model-predicted label (i.e., *Faithfulness*) and also provides a higher level of *Informativeness*. The human evaluation results are shown in Table 6. It can be observed that HINT gives the best results among all criteria.

6.3.3 Completeness and Sufficiency on ERASER. We also use the Evaluating Rationales And Simple English Reasoning (ERASER) benchmark (DeYoung et al. 2020) to evaluate model interpretability. ERASER contains seven datasets that are repurposed from existing NLP corpora originally used for sentiment analysis, natural language inference, question answering, and so forth. Each dataset is augmented with human annotated rationales (supporting evidence) that support output predictions. We select Movie Reviews[12] as our evaluation dataset. In ERASER, Movie Reviews only contains a total of 1,600 documents; another 200 test samples have been annotated with human rationales that are text spans indicative of the document polarity labels. We train all the models on our IMDB dataset and evaluate on the annotated Movie Reviews.

Following what has been proposed in ERASER, we first evaluate model interpretation using the two metrics, *Completeness* and *Sufficiency*, which measure the model prediction changes after removing the important words identified by the model and merely based on the important words,[13] respectively. That is:

$$\text{completeness} = m(x_i)_j - m(x_i/e_i)_j \tag{23}$$

$$\text{sufficiency} = m(x_i)_j - m(e_i)_j \tag{24}$$

12 http://www.eraserbenchmark.com/zipped/movies.tar.gz.

13 For HINT, we select the important words according to their weights from both the context representation learning and the topic representation modules.

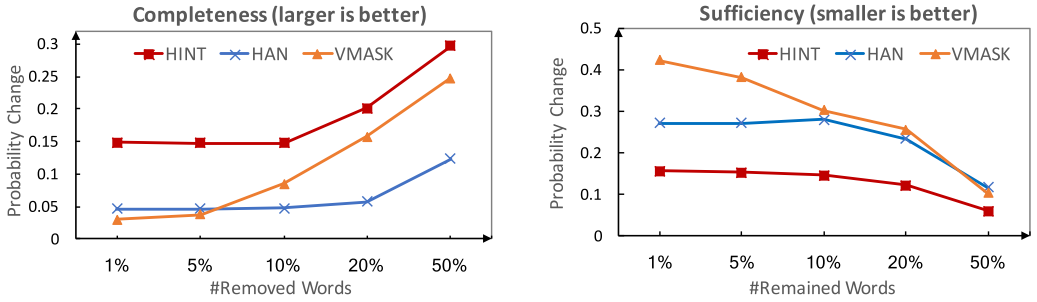


Figure 9
The *Completeness* and *Sufficiency* values by removing or keeping the top k% of most important word tokens identified by various models, $k \in \{1, 5, 10.20, 50\}$.

where x_i is the original text, e_i is the identified important words, and $m(\cdot)$ is the model probability on the predicted label j. To study the effects of word tokens in different importance level, we follow the setup in ERASER and group word tokens into 5 bins, each corresponding to the top 1%, 5%, 10%, 20%, and 50% of most important tokens identified by a model. The results are shown in Figure 9. We can observe that on *Completeness*, HAN and VMASK perform similarly, with up to 5% of most important words removed. But with more words removed, VMASK gives superior performance compared to HAN. HINT outperforms both HAN and VMASK consistently, but with the performance gap reduced when removing more words from documents. On *Sufficiency*, HINT and HAN give superior results compared to VMASK when only keeping a small number of important words. However, when over 20% of most important words are kept, the performance difference between HAN and VMASK diminishes. Overall, HINT gives the best results among all the models.

6.3.4 Agreement with Human Rationales. We use the multiple-aspect sentiment analysis dataset, BeerAdvocate (McAuley, Leskovec, and Jurafsky 2012), consisting of beer reviews, each of which is annotated with 5 aspects and the aspect-level rating scores in the range of 0 to 5. It has been widely used in evaluating rationale extraction models (Bastings, Aziz, and Titov 2019; Lei, Barzilay, and Jaakkola 2016; Li and Eisner 2019; Yu et al. 2021) by calculating the agreement between the annotated sentence-level rationales and model identified text spans. The common pipeline in much rationale extraction work is to predict binary masks for rationale selection, that is, masking the unimportant text spans and then predicting the sentiment scores only based on the selected rationales. In the HardKuma approach proposed for rationale extraction, constraints are further imposed to guarantee the continuity and sparsity of the selected text spans (Bastings, Aziz, and Titov 2019). More recently, Yu et al. (2021) argued that such a two-component pipeline approach tends to generate suboptimal results because even the first step of rationale selection selects a sub-optimal rationale; the sentiment predictor can still produce a lower prediction loss. To overcome this problem, they proposed the Attention-to-Rationale (A2R) approach by adding an additional predictor which predicts the sentiment scores based on soft attentions as opposed to the selected rationales. During training, the gap between the two predictors, one based on the selected rationales and the other based on soft attentions, is minimized. We show the

Table 7
Precision and recall of rationale extraction on the three aspects in the BeerAdvocate dataset. The results of HardKuma and A2R are taken from Yu et al. (2021).

	Look		Smell		Palate	
	Precision	Recall	Precision	Recall	Precision	Recall
HardKuma	81.0	69.9	74.0	**72.4**	45.4	**73.0**
A2R	**84.7**	**71.2**	**79.3**	71.3	64.2	60.9
VMASK	33.8	28.5	16.0	13.5	27.0	36.8
HAN	76.1	58.2	56.0	48.1	**71.6**	66.0
HINT	84.4	67.0	59.4	54.8	70.4	65.1

results of both HardKuma and A2R reported in Yu et al. (2021) in the upper part of Table 7.[14]

In our experiments, we train HINT and the baselines, VMASK and HAN, on the BeerAdvocate training set, and stop training when the models reach the smallest Mean Square Error on the validation set. Afterward, rationale selection is performed based on the word-level attention for VMASK (we use the top 15% of words as rationales), and based on the sentence-level importance scores for HAN and HINT. For the latter two models, we only extract the top sentence as the rationale for each document in the test set.

Following the setting in A2R (Yu et al. 2021), the overlap between the selected important words or sentences and the gold-standard rationales are calculated as precision and recall values and are shown in Table 7. It can be observed that approaches specifically designed for rationale extraction, HardKuma and A2R, give better results compared with other approaches that are not optimized for rationale extraction. VMASK performs the worst as it can only select token-level rationales. HINT outperforms HAN on both the *Look* and the *Smell* aspects by a large margin, and the two models give similar results on the *Palate* aspect.

6.4 Ablation Study

To study the effects of different modules in our model, we perform an ablation study and show the results in Table 8. In addition to the accuracy on the three datasets, we also report the interpretability metrics, namely, completeness and sufficiency for different variants.[15] For *Topic Representation Learning* (§3.1.2), we remove the Bayesian inference part that is used to learn word-level weight $\boldsymbol{\beta}_{ij}$. That is, rather than using $\boldsymbol{\beta}_{ij}$ to aggregate the word representations x_{ij} in order to derive the sentence embedding $\boldsymbol{r}_i$ as shown in Figure 3b, we now derive the sentence embedding $\boldsymbol{r}_i$ using the word-level TFIDF weights to aggregate the word representations x_{ij}. We also explore the effects without the regularization terms defined in Equations 10 and 17, respectively. Finally, we study the impact with or without the GATs and the number of GAT layers

14 Note that the HardKuma results reported in Yu et al. (2021) are inferior than those reported in the original paper (Bastings, Aziz, and Titov 2019). This is because the strong continuity constraint in HardKuma was not used in Yu et al. (2021) in order to achieve a fair comparison with A2R.

15 We randomly select 200 test samples to evaluate the interpretability.

Table 8
Ablation study results showing the effects of different input for topic learning reconstruction, regularization term, and GAT layers. Best accuracy and interpretability are marked **in bold**, the second best interpretablity is marked with underline. HINT achieves the overall best results on most metrics.

Methods	IMDB			Yelp			Guardian		
	Acc(↑)	Com(↑)	Suff(↓)	Acc(↑)	Com(↑)	Suff(↓)	Acc(↑)	Com(↑)	Suff(↓)
HINT	**89.11**	**0.21**	0.11	**98.52**	**0.22**	0.09	**95.37**	0.16	0.05
Remove Bayesian inference for β learning	89.02	0.17	0.14	98.45	0.16	0.11	95.21	0.16	0.06
Replace TFIDF with uniform weight	88.62	0.20	0.10	98.31	0.12	0.09	95.08	**0.18**	0.06
w/o the Regularization Term 1 (Equation (10))	89.03	0.18	0.11	98.49	0.11	**0.07**	95.20	0.15	0.07
w/o the Regularization Term 2 (Equation (17))	89.06	0.19	0.13	98.50	0.13	0.08	95.26	0.13	0.08
Remove GAT	89.00	0.17	0.10	98.41	**0.22**	0.10	94.87	0.08	**0.04**
2-layer GAT	88.93	0.17	0.11	98.52	0.18	0.10	94.99	0.14	0.06
4-layer GAT	88.85	0.18	**0.07**	98.50	0.18	0.09	93.58	0.13	**0.04**

in *Document Representation Learning* (§3.2). From the results in Table 8, HINT achieves best performance on accuracy and overall better performance on interpretability metrics. The variant of using uniform weight as an initialization for topic learning shows good interpretability on IMDB. This shows that with our proposed stochastic learning process for topic-related weights, it does not matter whether the word token weights are initialized by TFIDF or a uniform distribution. Although using multiple GAT layers fails to bring improvement to classification accuracy, 4-layer GAT has overall better interpretability performance than other GAT configurations.

6.5 Case Study

To show the capability of HINT in dealing with documents with mixed sentiments, we select one document from the IMDB dataset to illustrate the interpretations generated in Figure 10. The figure consists of three parts. The top part shows the word-level interpretations in the form of label-dependent words (in yellow) and label-independent words (in blue), as well as the sentence-level sentiment labels (sentence IDs highlighted with red or green colors). The middle part shows a heat map illustrating the association strengths between sentences and topics with a darker value indicating a stronger association. The lower part presents a bar chart showing the sentiment strength of each topic with the green and the red color for the positive and the negative sentiment, respectively. To make it easier to understand what each topic is about, we automatically extract the most relevant text span in the document to represent each topic (shown under the bar chart) by the approach described in Section 4.

Our model derives the document label by aggregating the sentence-level context representations weighted by their topic similarities (see in §3.2). From Figure 10, we can observe that Topic 1 appears to be the most prominent topic in the document from the sentence-topic heat map. Sentence S2 is related to Topic 2, while both sentences S3 and S4 are grouped under Topic 3. Among the three topics, Topic 1 is positive, while Topics 2 and 3 are negative. After aggregating sentences weighted by their topic similarities, the model infers an overall positive sentiment since the most prominent Topic 1 is positive. This example shows that HINT is able to capture both the topic and sentiment changes in text.

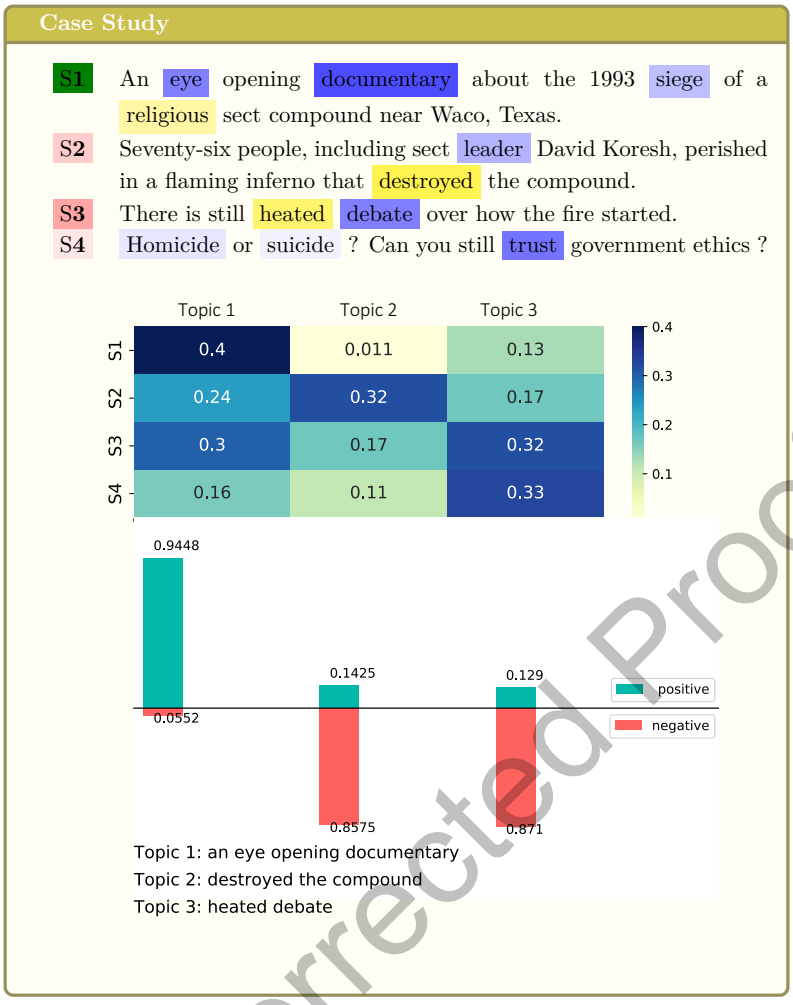


Figure 10
The upper part shows the document content with word-level important words and sentence-level sentiment labels. The middle and lower parts show the topic-related interpretations generated by HINT. The heat map shows the sentence-topic associations with a darker value indicating a stronger association, while the bar chart shows the sentiment strengths of the topics with the green and the red colors for the positive and the negative sentiment, respectively. We also display the topic labels (shown under the bar chart) by automatically extracting the most relevant text span in the document to represent each topic.

7. Conclusion and Future Work

In this article, we have proposed a Hierarchical Interpretable Neural Text classifier, called HINT, which automatically generates hierarchical interpretations of text classification results. It learns the sentence-level context and topic representations in an orthogonal manner in which the former captures the label-dependent semantic information while the latter encodes the label-independent topic information shared across documents. The learned sentence representations are subsequently aggregated by a

Graph Attention Network to derive the document-level representation for text classification. We have evaluated HINT on both review data and news data and shown that it achieves text classification performance on par with the existing neural text classifiers and generates more faithful interpretations as verified by both quantitative and qualitative evaluations.

Although we only focus on interpreting neural text classifiers here, the proposed framework can be extended to deal with other tasks such as content-based recommendation. In such a setup, we will need to learn both user- and item-based latent interest factors by analyzing reviews written by users and those associated with particular products. Because the proposed HINT is able to extract topics and their associated polarity strengths from reviews, it is possible to derive user- and item-based latent interest factors based on the outputs produced by HINT. Moreover, many NLP tasks such as natural language inference, rumor veracity classification, extractive question answering, and information extraction can be framed as classification problems. The proposed framework has a great potential to be extended to a wide range of NLP tasks.

Appendix A. Human Evaluation Instruction

HINT, HAN, and VAMSK generate different forms of interpretations. HAN can generate the interpretations based on the attention weights at both the word-level and the sentence-level. VMASK can only generate the interpretations at the word-level. Apart from the word-level and the sentence-level interpretations, HINT can also generate interpretations at the document-level by partitioning sentences into various topics and associating with each topic a polarity label. In the actual evaluation, to reduce cognitive load, we only present the most prominent topic in the document and the most contrastive topic in the form of word clouds to the evaluators. To retrieve the most prominent topic, we first identify the topic dimension with the largest value in the latent topic vector for each sentence and then select the most common topic dimension among all sentences. To select the most contrastive topic, we choose the one that has the minimal similarity with the first chosen topic. To generate the word cloud, we retrieve the topic words following the approach discussed in Section 4 with the vocabulary constrained to the local document. The evaluation schema is shown below:

Evaluation Schema

- **Correctness** – it measures to what extend the model-generated interpretation could lead to correct prediction. We present the interpretations generated by a model, and ask an evaluator to predict the document label based solely on the interpretations and check if the predicted label agrees with the **ground-truth label**.
- **Faithfulness** – it measures to what extend the interpretation generated is faithful to the model prediction. The evaluators should check if the interpretation generated will lead to the **model predicted label**.
- **Informativeness** – it measures to what extend the interpretation reveals the key information conveyed in text. We present the identified important words from HAN and VMASK; sentence importance scores from HAN and HINT. Additionally, we present the topic word clouds from HINT. We then ask users to evaluate the following aspects:
  - I know what the main topic is;
  - I can easily tell the polarity of the main topic;
  - I know what the secondary topic is (if there is any).

  We use the 1-5 likert scale (**strongly disagree, disagree, neutral, agree, strongly agree**) for each of the criteria above.

Appendix B. List of keywords used for Yelp reviews retrieval

The list of keywords used for retrieving patient reviews from Yelp is shown in Table A1.

Appendix C. Model Architecture and Parameter Setting

Our model architecture is shown in Table A2. We describe the parameter setup for each part of the model below:

Table A1
Keywords used to retrieve patient reviews from Yelp.

Walk-in Clinics, Surgeons, Oncologist, Cardiologists, Hospitals, Internal Medicine, Assisted Living Facilities, Cannabis Dispensaries, Doctors, Home Health Care, Health Coach, Emergency Pet Hospital, Pharmacy, Sleep Specialists, Professional Services, Addiction Medicine, Weight Loss Centers, Pediatric Dentists, Cosmetic Surgeons, Nephrologists, Naturopathic, Holistic, Pediatricians, Nurse Practitioner, Urgent Care, Orthopedists, Drugstores, Optometrists, Rehabilitation Center, Hypnosis, Hypnotherapy, Physical Therapy, Neurologist, Memory Care, Allergists, Counseling & Mental Health, Pet Groomers, Podiatrists, Dermatologists, Diagnostic Services, Radiologists, Medical Centers, Gastroenterologist, Obstetricians & Gynecologists, Pulmonologist, Ear Nose & Throat, Ophthalmologists, Sports Medicine, Nutritionists, Psychiatrists, Vascular Medicine, Cannabis Clinics, Hospice, First Aid Classes, Medical Spas, Spine Surgeons, Health Retreats, Medical Transportation, Dentists, Health & Medical, Speech Therapists, Emergency Medicine, Chiropractors, Medical Supplies, General Dentistry, Occupational Therapy, Urologists

Table A2
Model architecture.

Input:	A document d consists of M_d sentences $\{s_i\}_{i=1}^{M_d}$, $s_i = \{x_{ij}\}_{j=1}^{L}$	
Word Emb	Initialized by the GloVe embedding, $\{\boldsymbol{x}_{ij}\}_{j=1}^{L} \in \mathbb{R}^{N\times L}$	
Context learn	Word-level biLSTM	$\{\boldsymbol{x}_{ij}\}_{j=1}^{L} - \{\text{biLSTM}\} \rightarrow: \{\boldsymbol{h}_{ij}\}_{j=1}^{L} \in \mathbb{R}^{N\times L}$
	Attention layer	$\{\boldsymbol{h}_{ij}\}_{j=1}^{L} - \{\text{Linear}_1\} - \{\text{Linear}_2\} \rightarrow \alpha_{\text{ij}} \in \mathbb{R}^{\text{L}}$
	Context aggregate	$\sum_{j=1}^{L}\{\boldsymbol{h}_{ij}\}\alpha_{ij} \rightarrow \boldsymbol{s}_i \in \mathbb{R}^{N}$.
Topic learn	Word Weight Init.	$\boldsymbol{x}_i = \sum_{j=1}^{L} \text{TFIDF}_{ij} \cdot \boldsymbol{x}_{ij}$
	Bayesian inference	$\boldsymbol{x}_i - \{\text{Encoder}_1\} \rightarrow \boldsymbol{\mu}_\omega \in \mathbb{R}^{\text{N}}$ $\boldsymbol{x}_i - \{\text{Encoder}_2\} \rightarrow \log\boldsymbol{\sigma}_\omega^2 \in \mathbb{R}^{\text{N}}$ $\boldsymbol{\omega} = \text{Softmax}(\mu_\omega + \sigma_\omega \cdot \epsilon)$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
	Autorenoder	$\boldsymbol{\beta}^{1\times L} = \text{softmax}(\text{ReLU}(\boldsymbol{\omega}^{1\times d} \cdot \boldsymbol{x}_i)$ $\boldsymbol{r}_i = \sum_{j=1}^{L} \beta_{ij} \cdot \boldsymbol{x}_{ij}$ $\boldsymbol{z}_i = \text{softmax}(W_c \cdot \boldsymbol{x}_i + b_c)$ $\boldsymbol{r}_i' = \tanh(W_c' \cdot \boldsymbol{z}_i + b_c')$
Doc Modeling	Node Init.	$\boldsymbol{s}_i - \{\text{Linear3}\} \rightarrow \{\text{Linear4}\} \rightarrow \mathbf{s}_i^0$
	Edge weight init.	$e_{ij} = \text{softmax}(\mathbf{z}_i^\intercal \mathbf{z}_j)$
	Node update	$\mathbf{s}_i^{\ell+1} = \sigma(\sum_{j\in\mathcal{N}_i} e_{ij}\mathbf{W}s_j^\ell)$
		$\boldsymbol{d} = (\mathbf{s}_1^L + \mathbf{s}_2^L ... + \mathbf{s}_{M_d}^L)/M_d$
Classification	$\hat{y} = \text{softmax}\Big(\text{Linear}_6\big(\text{LeakyReLU}(\text{Linear}_5(\boldsymbol{d}))\big)\Big)$	

- *Context Learning* We use the pretrained 300-dimension GloVe embeddings with the dimension $N = 300$. The dimension of the word-level biLSTM hidden states is 150, and the dimension of the output $\boldsymbol{x}$ is also 300. The word embedding sequence is fed to two consecutive linear layers to obtain the attention weights. The weight matrices for the linear layers, Linear_1 and Linear_2, are $(300, 200)$ and $(200, 1)$, respectively. Then we aggregate $\hat{\boldsymbol{x}}$ by attention weights to obtain the sentence-level contextual representation $\boldsymbol{s}_i$.
- *Topic Learning* We calculate the TFIDF values for words offline. During inference, the TFIDF value of the out-of-vocabulary words is set to $1e-4$. For each sentence, we first normalize the TFIDF values of its constituent words and then aggregate the word embeddings weighted by their respective TFIDF values. This gives an initial sentence representation $\boldsymbol{x}_i \in \mathbb{R}^N$, which is then fed into two MLPs to generate the mean $\boldsymbol{\mu}_\omega$ and the variance $\log \sigma^2_\omega$, which are used to generate the output latent variable $\boldsymbol{\omega} \in \mathbb{R}^N$. After non-linear (`ReLU`) transformation and normalization (`Softmax`), we obtain the topic-aware weights $\boldsymbol{\beta}$ that is used to generate the input p_i for the autoenoder. The encoder and decoder in our autoencoder are 1-layer MLP with non-linear transformation. The weight matrices $\boldsymbol{W}_c$ and $\boldsymbol{W}'_c$ are $(300, K)$ and $(K, 300)$ respectively. K is the number of pre-defined topics. We set $K = 50$ for the two review datasets and $K = 30$ for the Guardian News data empirically.
- *Document Modeling* Graph nodes are initialized by the linear-transformed contextual sentence-level representations. The weight matrix in Linear_3 and Linear_4 are $(300, 200)$ and $(200, 50)$. The graph node dimension is 50.
- *Classification* The Linear_5 and Linear_6 have the dimensions of $(300, 200)$ and $(200, \#\text{labels})$, respectively.

We use dropout layers to alleviate overfitting, and insert a dropout layer after the word embedding layer, the word-level biLSTM layer, and after obtaining $\mathbf{z}_i$ and $\boldsymbol{\omega}$, respectively. The dropout rate is 0.4. We use the Adam (Kingma and Ba 2015) optimizer and set the learning rate to $1e-4$. The λ_1 and λ_2 in the regularization term are set to 0.05 and 0.01, respectively. η_a and η_b are set to 0.001 and 1, respectively. We train the model for 30 epochs and evaluate the performance at the end of each epoch. We report the average results for running 5 times with random seeds.

Acknowledgments

This work was funded by the UK Engineering and Physical Sciences Research Council (grant no. EP/T017112/1, EP/V048597/1, EP/X019063/1). Hanqi Yan receives the PhD scholarship funded jointly by the University of Warwick and the Chinese Scholarship Council. Yulan He is supported by a Turing AI Fellowship funded by the UK Research and Innovation (grant no. EP/V020579/1).

References

Abdou, Mostafa, Vinit Ravishankar, Maria Barrett, Yonatan Belinkov, Desmond Elliott, and Anders Søgaard. 2020. The sensitivity of language models and humans to Winograd schema perturbations. In *ACL*, pages 7590–7604. https://doi.org/10.18653/v1/2020.acl-main.679

Alvarez-Melis, David and Tommi S. Jaakkola. 2018. Towards robust

interpretability with self-explaining neural networks. In *NIPS*, pages 7786–7795.

Arnold, Sebastian, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A. Gers, and Alexander Löser. 2019. SECTOR: A neural model for coherent topic segmentation and classification. *Transactions of the Association for Computational Linguistics*, 7:169–184. https://doi.org/10.1162/tacl_a_00261

Bang, Seo-Jin, Pengtao Xie, Heewook Lee, Wei Wu, and Eric P. Xing. 2021. Explaining a black-box by using a deep variational information bottleneck approach. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021*, pages 11396–11404. https://doi.org/10.1609/aaai.v35i13.17358

Bastings, Jasmijn, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers*, pages 2963–2977. https://doi.org/10.18653/v1/P19-1284

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *JMLR*, 3:993–1022.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NIPS*, pages 1877–1901.

Card, Dallas, Chenhao Tan, and Noah A. Smith. 2018. Neural models for documents with metadata. In *ACL*, pages 2031–2040. https://doi.org/10.18653/v1/P18-1189

Chaney, Allison and David Blei. 2021. Visualizing topic models. *Proceedings of the International AAAI Conference on Web and Social Media*, 6(1):419–422.

Chen, Hanjie and Yangfeng Ji. 2020. Learning variational word masks to improve the interpretability of neural text classifiers. In *EMNLP*, pages 4236–4251. https://doi.org/10.18653/v1/2020.emnlp-main.347

Chen, Hanjie, Guangtao Zheng, and Yangfeng Ji. 2020. Generating hierarchical explanations on text classification via feature interaction detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5578–5593. https://doi.org/10.18653/v1/2020.acl-main.494

Chen, Jianbo, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 882–891.

Chen, Jun, Xiaoya Dai, Quan Yuan, Chao Lu, and Haifeng Huang. 2020. Towards interpretable clinical diagnosis with Bayesian Network Ensembles stacked on entity-aware CNNs. In *ACL*, pages 3143–3153. https://doi.org/10.18653/v1/2020.acl-main.286

De-Arteaga, Maria, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128. https://doi.org/10.1145/3287560.3287572

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

DeYoung, Jay, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 4443–4458. https://doi.org/10.18653/v1/2020.acl-main.408

Guan, Chaoyu, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. 2019. Towards a deep and unified understanding of deep neural models in NLP. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 2454–2463.

Gui, Lin and Yulan He. 2021. Understanding patient reviews with minimum

supervision. *Artificial Intelligence in Medicine*, 120:102160. https://doi.org/10.1016/j.artmed.2021.102160, PubMed: 34629148

Gui, Lin, Jia Leng, Jiyun Zhou, Ruifeng Xu, and Yulan He. 2022. Multi task mutual learning for joint sentiment classification and topic detection. *IEEE Transactions on Knowledge and Data Engineering*, 34(4):1915–1927. https://doi.org/10.1109/TKDE.2020.2999489

Jacovi, Alon and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *ACL*, pages 4198–4205. https://doi.org/10.18653/v1/2020.acl-main.386

Jain, Sarthak and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 3543–3556. https://doi.org/10.18653/v1/N19-1357

Jawahar, Ganesh, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *ACL*, pages 3651–3657. https://doi.org/10.18653/v1/P19-1356

Jiang, Chengyue, Yinggong Zhao, Shanbo Chu, Libin Shen, and Kewei Tu. 2020. Cold-start and interpretability: Turning regular expressions into trainable recurrent neural networks. In *EMNLP*, pages 3193–3207. https://doi.org/10.18653/v1/2020.emnlp-main.258

Jin, Xisen, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2020. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. In *8th International Conference on Learning Representations, ICLR 2020*, OpenReview.net.

Johansson, Fredrik D., Uri Shalit, and David A. Sontag. 2016. Learning representations for counterfactual inference. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 3020–3029, JMLR.org.

Kim, Siwon, Jihun Yi, Eunji Kim, and Sungroh Yoon. 2020. Interpretation of NLP models through input marginalization. In *EMNLP*, pages 3154–3167. https://doi.org/10.18653/v1/2020.emnlp-main.255

Kingma, Diederik P. and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*.

Kingma, Diederik P. and Max Welling. 2014. Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings*.

Lai, Vivian and Chenhao Tan. 2019. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 29–38. https://doi.org/10.1145/3287560.3287590

Lei, Tao, Regina Barzilay, and Tommi S. Jaakkola. 2016. Rationalizing neural predictions. In *EMNLP*, pages 107–117. https://doi.org/10.18653/v1/D16-1011

Li, Jiwei, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220.

Li, Xiang Lisa and Jason Eisner. 2019. Specializing word embeddings (for parsing) by information bottleneck. In *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*, pages 2744–2754. https://doi.org/10.18653/v1/D19-1276

Lin, Chenghua, Yulan He, Richard Everson, and Stefan Ruger. 2012. Weakly supervised joint sentiment-topic detection from text. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):1134–1145. https://doi.org/10.1109/TKDE.2011.48

Lipton, Zachary C. 2018. The mythos of model interpretability. *Communications of the ACM*, 61(10):36–43. https://doi.org/10.1145/3233231

Maas, Andrew, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL-HLT*, pages 142–150.

McAuley, Julian, Jure Leskovec, and Dan Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. In *2012 IEEE 12th International Conference on Data Mining*, pages 1020–1025.

Niu, Xing, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. 2020. Evaluating robustness to input perturbations for neural machine translation. In *ACL*,

pages 8538–8544. https://doi.org/10.18653/v1/2020.acl-main.755

O'Hare, Neil, Michael Davy, Adam Bermingham, Paul Ferguson, Páraic Sheridan, Cathal Gurrin, and Alan F. Smeaton. 2009. Topic-dependent sentiment analysis of financial blogs. In *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion, TSA '09*, pages 9–16. https://doi.org/10.1145/1651461.1651464

Pruthi, Danish, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 4782–4793. https://doi.org/10.18653/v1/2020.acl-main.432

Ribeiro, Marco Túlio, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with checklist. In *ACL*, pages 4902–4912. https://doi.org/10.18653/v1/2020.acl-main.442

Rieger, Laura, Chandan Singh, William Murdoch, and Bin Yu. 2020. Interpretations are useful: Penalizing explanations to align neural networks with prior knowledge. In *ICML*, pages 8116–8126.

Selvaraju, Ramprasaath R., Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2020. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *IJCV*, 128(2):336–359. https://doi.org/10.1007/s11263-019-01228-7

Serrano, Sofia and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers*, pages 2931–2951. https://doi.org/10.18653/v1/P19-1282

Singh, Chandan, W. James Murdoch, and Bin Yu. 2019. Hierarchical interpretations for neural network predictions. In *ICLR*.

Tang, Zheng, Gus Hahn-Powell, and Mihai Surdeanu. 2020. Exploring interpretability in event extraction: Multitask learning of a neural event classifier and an explanation decoder. In *ACL*, pages 169–175. https://doi.org/10.18653/v1/2020.acl-srw.23

Wang, Zhengjue, Chaojie Wang, Hao Zhang, Zhibin Duan, Mingyuan Zhou, and Bo Chen. 2020. Learning dynamic hierarchical topic graph with graph convolutional network for document classification. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3959–3969.

Wiegreffe, Sarah and Yuval Pinter. 2019. Attention is not not explanation. In *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*, pages 11–20. https://doi.org/10.18653/v1/D19-1002

Wu, Zhiyong, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *ACL*, pages 4166–4176. https://doi.org/10.18653/v1/2020.acl-main.383

Xie, Qianqian, Jimin Huang, Pan Du, Min Peng, and Jian-Yun Nie. 2021. Graph topic neural network for document representation. In *Proceedings of the Web Conference 2021*, WWW '21, pages 3055–3065. https://doi.org/10.1145/3442381.3450045

Yan, Hanqi, Lin Gui, Gabriele Pergola, and Yulan He. 2021. Position bias mitigation: A knowledge-aware graph model for emotion cause extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers)*, pages 3364–3375. https://doi.org/10.18653/v1/2021.acl-long.261

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *NIPS*, pages 5754–5764.

Yang, Zichao, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL*, pages 1480–1489. https://doi.org/10.18653/v1/N16-1174

Yu, Mo, Yang Zhang, Shiyu Chang, and Tommi S. Jaakkola. 2021. Understanding interlocking dynamics of cooperative rationalization. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pages 12822–12835.

Zanzotto, Fabio Massimo, Andrea Santilli, Leonardo Ranaldi, Dario Onorati, Pierfrancesco Tommasino, and Francesca Fallucchi. 2020. KERMIT: Complementing

transformer architectures with encoders of explicit syntactic interpretations. In *EMNLP*, pages 256–267. https://doi.org/10.18653/v1/2020.emnlp-main.18

Zhang, Jingyuan, Mingming Sun, Yue Feng, and Ping Li. 2020. Learning interpretable relationships between entities, relations and concepts via Bayesian structure learning on open domain facts. In *ACL*, pages 8045–8056. https://doi.org/10.18653/v1/2020.acl-main.717

Zhou, Fan, Shengming Zhang, and Yi Yang. 2020. Interpretable operational risk classification with semi-supervised variational autoencoder. In *ACL*, pages 846–852. https://doi.org/10.18653/v1/2020.acl-main.78

Zhou, Wangchunshu, Jinyi Hu, Hanlin Zhang, Xiaodan Liang, Maosong Sun, Chenyan Xiong, and Jian Tang. 2020. Towards interpretable natural language understanding with explanations as latent variables. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, pages 6803–6814.