AN INTERPRETABLE CONTRASTIVE LOGICAL KNOWLEDGE LEARNING METHOD FOR SENTIMENT ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Current interpretable sentiment analysis (ISA) methods frequently underperform state-of-the-art models, and few of them cast light on the inner working of pre-trained models. In this work, we fill the gap by addressing four key research challenges in ISA—knowledge acquisition, knowledge representation, knowledge learning and knowledge reasoning—in one unified framework. Theoretically, we propose a novel contrasitive logical knowledge learning (CLK) framework that can visualize the decisions made through deterministic Talmudic public announcement logic semantics. We apply CLK to current popular sentiment analysis models to obtain CLK based interpretable ones. Empirically, experimental results of both binary sentiment analysis tasks and fine-grained sentiment analysis tasks indicate that CLK can achieve an effective trade-off between accuracy and interpretability. Furthermore, we find that CLK can reduce the uncertainty of logical knowledge for discriminative labels in unbalanced fine-grained sentiment analysis tasks. Besides, we carry out a case study to investigate the fidelity of model interpretability through knowledge reasoning, which demonstrates that the explanations provided by our method are causally effective for sentiment analysis tasks.

1 INTRODUCTION

Sentiment analysis (also known as opinion mining) is a primary task in natural language processing that identifies the emotional inclination of humans toward particular topics (Zitnik et al., 2022), it is commonly used in marketing (Valle-Cruz et al., 2022), customer service (Li et al., 2019), medical healthcare (Sanglerdsinlapachai et al., 2021), and social media (Arias et al., 2022). Investigating model interpretability is critical to obtain user trust in high-stake domains (Ito et al., 2020; Zhang et al., 2013). Several approaches have been proposed for interpretability in sentiment analysis, such as Tsetlin Machine (Yadav et al., 2021), Hidden Markov Model based methods (Perikos et al., 2021), and fuzzy rule based methods (Liu & Cocea, 2017). However, while the existing approaches can provide explanations, the fidelity and rationality of model interpretability are unclear for sentiment analysis tasks, which is insufficient for user trust. In addition, despite that deep learning models have shown promising performance (Alaparthi & Mishra, 2020), few interpretable methods have involved them in sentiment analysis tasks.

According to (Chen et al., 2022), the logical knowledge implicitly in bidirectional recurrent neural networks (BRNNs) can be represented by Talmudic public announcement logic (TPK) (Abraham et al., 2013), since TPK has natural temporal characteristics and can model context dependency. In light of this, it's possible to learn the interpretable TPK from bidirectional deep neural networks (BDNNs) by modeling word-label dependency, since the word-label dependency in sequential data can also be represented by temporal logic. Considering the sequence "I like this novel very much, but the film is terrible, so I will give a bad review", label semantics are associated with correlated words in the sequence. Specifically, the label "positive" is associated with the words "like", and the label "negative" is associated with the words "terrible" and "bad", the predicted label is determined by the temporal characteristics of these correlated words. A few studies have looked into the correlation between words and labels in sentiment analysis (Ito et al., 2020), the logical knowledge implicitly in the trained deep learning models is still unspecified although it can help us to reveal how a BDNN operates. It is worth noting that the knowledge representation of deep learning models is challenging due to the anisotropy and uncertainty of logical knowledge, where anisotropy is produced by the vector representation derived from transformer when using pre-trained models, meaning that the vectors will be unevenly spaced and filled in a small conical area (Cai et al., 2020; Ethayarajh, 2019). The uncertainty stems from two avenues. On the one hand, the knowledge of labels with similar meanings (like ”slightly positive” and ”positive”) has similar logic knowledge, which is easily confused; on the other hand, logical knowledge trained with model during the training stage leads to randomness. As a result, the following research questions emerge naturally to fill the research gap:(1) *how to acquire effective logical knowledge and integrate it into model training in sentiment analysis tasks?*(2) *how to generate human-understandable explanations for sentiment analysis tasks and evaluate the fidelity of model interpretability?*

To answer the first research question, we propose a contrastive logical knowledge learning (CLK) method based on the Talmudic public announcement logic (TPK) and contrastive learning. CLK can be regarded as a component that can be added to any deep learning models, and the challenge is that models constructed in such a way may suffer from lower performance when the transparency is brought in, thus the logical knowledge need to be accurate and effective. To address this challenge, we design a knowledge acquisition module (Figure 1 (a)), a knowledge representation module (Figure 1 (b)) and a knowledge learning module (Figure 1 (c)) in CLK. The logical knowledge is obtained from two avenues: firstly, we employ the latent Dirichlet allocation to extract the semantic knowledge from existing datasets, and further utilize this logical knowledge to form a good label representation (Section 3.1). Secondly, we learn the potential public announcements of TPK by calculating the similarity of label representation and word representation (Section 3.3). Based on the learned public announcements, the logical knowledge can be represented in the form of TPK (Section 3.2). To address the anisotropy and uncertainty of logical knowledge, we aggregate it with contrastive learning to enhance the uniformity and reduce uncertainty, thus encouraging the model to learn more discriminative representations for similar labels (Section 3.3).

To answer the second question, we design a knowledge reasoning module in CLK (Section 3.4), which can generate human-readable explanations based on TPK semantics and validate the fidelity of the model interpretability by causal reasoning. By displaying how sentiment changes from the beginning to the end of sequences, CLK can visualize decisions made in the form of TPK models. Through reasoning on a structural causal model for sentiment analysis tasks, the effectiveness and rationality of the model interpretability are investigated. In addition, we conduct extensive experiments on both binary sentiment analysis datasets and fine-grained sentiment analysis datasets, and the results demonstrate that CLK can achieve an effective trade-off between accuracy and interpretability by leveraging the benefits of both logical knowledge and contrastive learning, especially compared with pre-trained models on Weibo-8 and Yelp-2 dataset (Section 5). Besides, we conduct an ablation study to show how knowledge acquisition module and knowledge learning module influence model output (Section 5.3).

2 Deterministic Talmudic Public Announcement logic

In this section, we briefly introduce the syntax and semantics of of deterministic Talmudic Public Announcement logic (TPK), and the detailed background of TPK can be found in Appendix B.

Syntax The syntax of TPK is inspired by modal logic and public announcement logic. Fix the non-empty sets of propositions and modal operators $\Box$, $\boxminus$, Y and $\mathbb{Y}$, the minimal syntax of TPK can be specified in Backus–Naur form as follows:

$$\phi ::= p_i \mid \neg\phi \mid \phi \vee \psi \mid \phi \wedge \psi \mid \Box\phi \mid \boxminus \phi \mid Y\phi \mid \mathbb{Y}\phi \tag{1}$$

where p_i is a propositional letter for $i \in \mathbb{N}$, ϕ and ψ are well-formed TPK formulas. Notice that $\Box$ is the standard necessity operator, Y is a yesterday operator for $\Box$, $\boxminus$ is a modal operator corresponding to function ρ, and $\mathbb{Y}$ is a yesterday operator for $\boxminus$. ρ is a functional relation that denotes the public announcement in TPK, which is specified as below.

Definition Based on the above language of TPK, *a deterministic TPK (Taimudic public announcement logic) model (Abraham et al., 2013) is defined as a 6-tuple $\langle S, R, R^*, \rho, s_0, \pi\rangle$ where (S, R, s_0) is a directed tree with root s_0 and tree successor relation R, R^* is the transitive closure of R, ρ: $S \to S$ is a functional relation, π is the valuation function.*

For $\forall x, y, z, v \in S$, we say ρ is a public announcement of TPK iff ρ satisfies the following properties:(a) $x\rho y \to y \neq s_0$. (b) Let zRx, the successors of z are publicly clarified to be v, where zRv holds, if we have $x\rho y$, with vR^*y. (c) For deterministic actions, $x\rho y \wedge x\rho z \to y = z$. (d) Since every z has a unique successor which is publicly clarified at some point, this means that there exists a path the model deterministic. (e) Let D be the distance from the root and $u, v \in S$, whenever uRv then $D(v) = D(u) + 1$. $x\rho y \wedge \neg(yRx) \wedge y \neq x \rightarrow D(y) = D(x)$ [1].

Semantics Formulas of TPK are interpreted over state transition systems whose transitions are determined by the non-deterministic (deterministic) actions of agents. In TPK, the modal operators can describe the property of states i.e., the path formula $s \vDash \Box\phi$ is read "ϕ holds in each state that is accessible to s ". For $\forall t, s \in S$, the semantics of TPK are as follows: as for the relation R: $t \vDash \Box A$ iff for all s such that tRs we have $s \vDash A$; $t \vDash YA$ iff for all s such that sRt we have $s \vDash A$; If $t = s_0$ then $YA = \perp$. As for the functional relation ρ: $t \vDash \boxminus A$ iff for all s such that $t\rho s$ we have $s \vDash A$; $t \vDash \mathbb{Y}A$ iff for all s such that $s\rho t$ we have $s \models A$ and if no such s exists then $\perp$.

Now we can write axioms and attempt to prove completeness based on above operators, but it is not our purpose in this paper, the interested reader can find more details in (Abraham et al., 2013).

3 METHOD

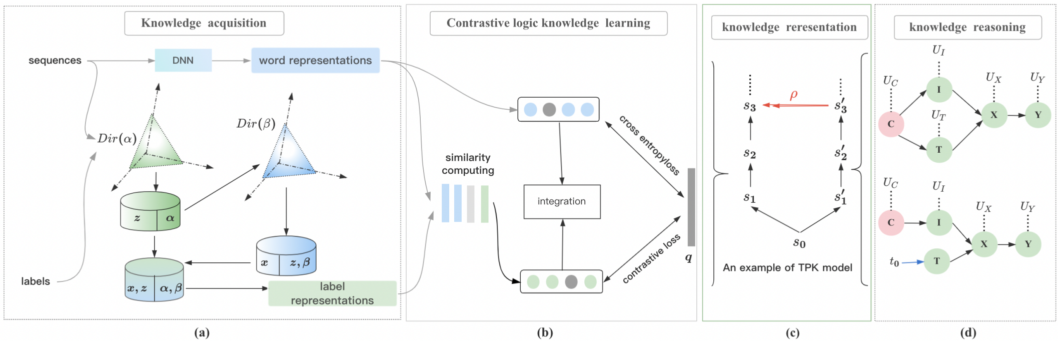


Figure 1: The overall framework of our proposed method. $Dir(\cdot)$ in (a) denotes a Dirichlet distribution; $\boldsymbol{q}$ in (b) denotes the ground truth; s_i $(i \geq 0)$ in (c) denotes the i-th state in TPK model; a green (or blue) circle in (d) indicates a variable while a pink circle indicates a confounder.

3.1 KNOWLEDGE ACQUISITION

Latent Dirichlet Allocation (LDA) is a topic model which can give the topic of each document in the document set in the form of a probability distribution (Blei et al., 2003). There is only one corresponding label for each input sequence in our sentiment analysis datasets. Firstly, we classify all sequences into documents based on their labels and obtain corresponding documents.Let $\mathbb{D}_i$ represent the sequence-level document of i-th label category, we have $\mathbb{D}_i = \{X_1, X_2, ...X_V\}$, where X_v denotes the v-th sequence in the document, V denotes the number of sequences labeled as $\mathbb{D}_i$ in the datasets, and N_v is the number of words in v-th sequence. By dividing each sequence into a series of tokens (words) and utilizing these tokens to represent the document, we obtain $D_i = \{x_1, x_2, ...x_M\}$, where M is the number of tokens and x_m indicates the m-th token, D_i denotes the token-level document of i-th label category. Suppose the document D_i has K $(K > 0)$ topics, the topic probability distribution of X_v is denoted as $\theta_v = P(\boldsymbol{z}|X_v)$, where $\boldsymbol{z} = [z_1, z_2, \ldots, z_k]$ is a topic vector. Thus the topic probability distribution of all sequences in D_i can be represented as $\boldsymbol{\Theta} = \{\boldsymbol{\theta_v}\}_{v=1}^{V} \in R^{V \times K}$, where the prior distribution of $\boldsymbol{\theta_v}$ is a Dirichlet distribution with prior parameter $\boldsymbol{\alpha}$ such that $P(\boldsymbol{\theta_v}|\boldsymbol{\alpha}) \sim Dirichlet(\boldsymbol{\theta_v}|\boldsymbol{\alpha})$. Assume that there are N_v words in sequence X_v, and let vector $\boldsymbol{n}_v = [n_v^1, n_v^2, \ldots, n_v^k]$ denote the number of occurrence of each topic in X_v, where n_v^k denotes the number of occurrence of topic z_k. Then $\boldsymbol{n}_v \sim multi(\boldsymbol{n}_v|\boldsymbol{\theta_v}, N_v)$, where $multi(\cdot)$ represents the multinomial distribution function. Since a Dirichlet distribution is the prior distribution of a multinomial distribution (see Appendix F.1), the posterior distribution of $\boldsymbol{\theta_v}$ can be computed by $P(\boldsymbol{\theta_v}|\boldsymbol{n_v}, \boldsymbol{\alpha}) \sim Dirichlet(\boldsymbol{\theta_v}|(\boldsymbol{\alpha} + \boldsymbol{n_v}))$. Since all sequences in document D_i are independent of each other, the probability of the topic in the corpus can be obtained by (see Appendix F.2 for more details):

$$P(\boldsymbol{z}|\boldsymbol{\alpha}) = \prod_{v=1}^{V} P(\boldsymbol{z_v}|\boldsymbol{\alpha}) = \prod_{v=1}^{V} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod\limits_{k=1}^{K} \Gamma(\alpha_k)} \frac{\prod\limits_{k=1}^{K} \Gamma(\alpha_k + n_v^k)}{\Gamma(\sum_{k=1}^{K}(\alpha_k + n_v^k))} \tag{2}$$

Let $\boldsymbol{\phi} = P(\boldsymbol{w}|z_k)$ represent the word probability distribution of z_k, then the word probability distribution of all topics is given by: $\Phi = \{\boldsymbol{\phi_k}\}_{k=1}^{K} \in R^{K \times M}$. Similarly, the prior distribution of $\boldsymbol{\phi_k}$

[1] TPK may appear to the perceived readers to be similar to temporal logic and traditional announcement logic, we will discuss their differences in Appendix E .

is given by a Dirichlet distribution with prior parameter $\boldsymbol{\beta}$ such that $P(\boldsymbol{\phi_k}|\boldsymbol{\beta}) \sim Dirichlet(\boldsymbol{\phi_k}|\boldsymbol{\beta})$. Assume that there are N_k words in topic z_k, and let vector $\boldsymbol{n}_k = [n_k^1, n_k^2, \ldots, n_k^m]$ denote the number of occurrence of each words in topic z_k, then $\boldsymbol{n}_k \sim multi(\boldsymbol{n}_k|\boldsymbol{\phi_k}, \boldsymbol{N_k})$. Similarly, the posterior distribution of ϕ_k is given by $P(\boldsymbol{\phi_k}|\boldsymbol{n_k}, \boldsymbol{\beta}) \sim Dirichlet(\boldsymbol{\phi_k}|(\boldsymbol{\beta} + \boldsymbol{n_k}))$. Since these topics in the document are independent of each other, the word probability distribution in D_i can be obtained as follows (Blei et al., 2003) (see Appendix F.3 for more details):

$$P(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\beta}) = \prod_{k=1}^{K} P(\boldsymbol{x_k}|\boldsymbol{\beta}) = \prod_{k=1}^{K} \frac{\Gamma(\sum_{m=1}^{M} \beta_m)}{\prod\limits_{m=1}^{M} \Gamma(\beta_m)} \frac{\prod\limits_{m=1}^{M} \Gamma(\beta_m + n_k^m)}{\Gamma(\sum_{m=1}^{M}(\beta_m + n_k^m))} \tag{3}$$

Above information allows us to derive the following formula for joint probability distribution of word-topic:

$$P(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{k=1}^{K} \frac{\Gamma(\sum_{m=1}^{M} \beta_m)}{\prod\limits_{m=1}^{M} \Gamma(\beta_m)} \frac{\prod\limits_{m=1}^{M} \Gamma(\beta_m + n_k^m)}{\Gamma(\sum_{m=1}^{M}(\beta_m + n_k^m))} \prod_{v=1}^{V} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod\limits_{k=1}^{K} \Gamma(\alpha_k)} \frac{\prod\limits_{k=1}^{K} \Gamma(\alpha_k + n_v^k)}{\Gamma(\sum_{k=1}^{K}(\alpha_k + n_v^k))} \tag{4}$$

where $P(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\alpha}, \boldsymbol{\beta})$ represents the topic-word probability. This paper sets $\alpha = 0.01$, $\beta = 0.01$ and utilizes variational inference-based EM algorithm (Wang et al., 2018) to produce the probability distributions, then the probability $P(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\alpha}, \boldsymbol{\beta})$ is solved iteratively. Based on the solved $P(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\alpha}, \boldsymbol{\beta})$, we can get the words with high probability under a certain topic by sorting the probability values. Since we have divided the training data into several documents according to their labels, we can directly extract T words with the highest probability from these documents based on $P(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\alpha}, \boldsymbol{\beta})$, then utilize them to represent the i-th label such that $D_i' = \{w_1, w_2, \ldots, w_T\}$.

Glove vectors is a word representation tool that considers the statistics of words (by co-occurrence matrix) in the corpus to learn word representation (Pennington et al., 2014). To obtain a better word representation for above T words, we introduce a pre-trained glove model provided by (Pennington et al., 2014), and get a set of word vectors $\boldsymbol{h}_{D_i'} = \{\boldsymbol{h}_{w_1}, \boldsymbol{h}_{w_2}, \ldots \boldsymbol{h}_{w_T}\}$, where $\boldsymbol{h}_{w_i}$ is the word vector of w_i based on the pre-trained Glove model. Remark that the total number of these vectors are T since we have extracted top T $(T \geq 0)$ words based on $P(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\alpha}, \boldsymbol{\beta})$. Then, the average of vectors of all w_i can represent D_i as $\boldsymbol{h}_{D_i} = \frac{1}{T}\sum_{j=1}^{T} \boldsymbol{h}_{w_j}$, where $\boldsymbol{h}_{D_i}$ denotes the average vector of D_i. This paper set $T = 25$ and use $\boldsymbol{h}_{D_i}$ it to initial label embeddings.

3.2 Knowledge representation

For each input sequence $X = (x_1, x_2, ..., x_n)$, we get the corresponding feature vectors (hidden states) $\boldsymbol{h_{e_i}}$ of each word by a base model (or baseline), denoted as $\boldsymbol{h} = [\boldsymbol{h}_{e_1}, \boldsymbol{h}_{e_2}, \ldots \boldsymbol{h}_{e_n}]$, where e_i denotes the text embedding vector of x_i by the Embedding layer and $\boldsymbol{h}_{e_i}$ denotes the hidden state of word x_i through a deep learning network. Then we employ two dense layers with different activation functions to get $\boldsymbol{p} = softmax(relu(\boldsymbol{h}))$, where function $relu(x) = max(0, x)$ and function $softmax(z_i) = \frac{e^{z_i}}{\sum_{i=1}^{k} e^{z_i}}$. The base model just computes the loss by $\boldsymbol{p}$ and true labels. Despite knowing this training process, we still know nothing about the logic relationship of the hidden states in the base model. Inspired by the work of (Chen et al., 2022), we employ the deterministic Talmudic public announcement logic (TPK) to represent this kind of logical knowledge. As stated in Section 2, TPK is composed of a 6-tuple $\langle S, R, R^*, \rho, s_0, \pi \rangle$. Thus, for the state set S of a TPK model, we construct them by the obtained sequence of DNNs such that $s_i = \{\boldsymbol{h}_{e_1}, ..., \boldsymbol{h}_{e_i}, r\} \quad (i > 0)$, where $s_i \in S$ denotes the i-th state; r denotes a propositional letter generated by the learned TPK; the root s_0 is an empty state and denotes the start of the model.

Similarly to (Chen et al., 2022), the connectivity of these states is represented by the successor relation R and transitive closure R^* in TPK. Public announcement function ρ in TPK can represent the logic relationship of hidden states by showing how current states are determined by future information. Notice that, this kind of logical relationship always changes during training, necessitating the training of the model with public announcements. In next subsection, we will discuss how to learn the public announcements of TPK and how to employ the logical knowledge to guide model training. The assignment function π can reflect the information distribution of each word (or sequence) since it stores the true value of each proposition variable. Let $\mathcal{Y}$ denote the information of each state, then $\pi = Distribution\{\mathcal{Y}\}$, where $Distribution\{\mathcal{Y}\}$ denotes the information distribution of $\mathcal{Y}$, which is established by $\mathbb{M}$ in next subsection.

3.3 CONTRASTIVE LOGICAL KNOWLEDGE LEARNING

The uncertainty of logical knowledge is the main obstacle in learning effective representations. Different from the work in (Chen et al., 2022), this paper learns potential public announcements through similarity computing and contrastive learning. Since the label vector $\boldsymbol{h}_{D_i}$ contains the logical knowledge from datasets, we use these $\boldsymbol{h}_{D_i}$ to initialize label embeddings and then get $\boldsymbol{e}^l = [\boldsymbol{e}^{D_1}, \boldsymbol{e}^{D_2}, ...\boldsymbol{e}^{D_n}](1 \leq n \leq N^l)$, where $\boldsymbol{e}^{D_i}$ denotes the embedding vector of i-th label. Notably, for a non-pretrained based model, we use a DNN (BLTM or BGRU) to encode each label vector after label embedding, and obtain the label vector $\boldsymbol{h}^l = [\boldsymbol{h}_{e^{D_1}}, \boldsymbol{h}_{e^{D_2}}, ...\boldsymbol{h}_{e^{D_n}}](1 \leq n \leq N^l)$. Then we calculate the similarity matrix $M \in \mathbb{R}^{n \times |N^l|}$ of each word vector $\boldsymbol{h}_{e_j} \in \boldsymbol{h}$ and label vector $\boldsymbol{h}_{e^{D_i}} \in \boldsymbol{h}^l$ in CLK as below:

$$\begin{cases} \mathbb{P}_{i,j} = h_{e^{D_i}} \cdot tanh(f_1 h_{e^{D_i}} + f_2 h_{e_j}) \\ \mathbb{M}_{i,j} = Softmax(\mathbb{P}_{i,j}) = \dfrac{exp(\mathbb{P}_{i,j})}{\sum_{j=1}^{n} exp(\mathbb{P}_{i,j})} \end{cases} \tag{5}$$

where f_1 and f_2 are trained parameters. Remark that, for pre-trained models, we directly use $\boldsymbol{e}^l$ and $\boldsymbol{h}$ to calculate the similarity of word vectors and label vectors since the model has been trained. The information of each state and public announcements of a sequence X_i is stored in $\mathbb{M}_{i,j}$. Since the public announcements change the decision of a TPK model, we can obtain them by comparing and analyzing the maximum values of elements in each row of $\mathbb{M}_{i,j}$. And a public announcement can be a word or a phrase (comprise of several words) that changes model's choices into a another direction based on $\mathbb{M}$. Theoretically, the logical knowledge inferred from datasets should match what is actually known. However, there are some distinctions between logical knowledge and real knowledge in embedding space because of uncertainty and anisotropy. To address this issue, CLK aggregates the logical knowledge with a contrastive term from the standpoint of distance, thus confining it to a space congruent with the real knowledge. Given that CLK may shorten the gap between pieces of similar knowledge in the embedding space through contrastive learning, a contrastive objective $\mathcal{L}_c$ is introduced to guide model training (Hadsell et al., 2006). To make use of the correlation information of the word-label pairs and local information from base model, the contrastive logical knowledge $\tilde{\boldsymbol{q}}_{\boldsymbol{i}}$ is computed by:

$$\tilde{\boldsymbol{q}}_{\boldsymbol{i}} = Sigmoid(M') = \frac{1}{1 + e^{-M'}} = \frac{1}{1 + e^{-Relu(\mathbb{M} \cup \boldsymbol{h})}} = \frac{1}{1 + e^{-max(0, \boldsymbol{w}^T(\mathbb{M} \cup \boldsymbol{h}) + b})} \tag{6}$$

where $\boldsymbol{w}^T$ and b are two trained parameters. Let $\boldsymbol{q}_i$ be the ground truth of X_i, then the contrastive objective $\mathcal{L}_c$ is given by:

$$\mathcal{L}_c(\boldsymbol{q}_i, \tilde{\boldsymbol{q}}_{\boldsymbol{i}}) = \tilde{\boldsymbol{q}}_{\boldsymbol{i}} \|\boldsymbol{q}_{\boldsymbol{i}}\| + (1 - \tilde{\boldsymbol{q}}_{\boldsymbol{i}}) * \| \max(margin - \boldsymbol{q}_{\boldsymbol{i}}), 0)\|) \tag{7}$$

$margin \in [0, 1]$ is a given parameter to control the strength of contrastive learning. Through the objective function $\mathcal{L}_c$, the semantic knowledge of similar labels in embedding space should be as close as possible, while the different ones should be as far as possible. As stated in (Wang & Isola, 2020), a good contrastive learning system possess both alignment and uniformity, while it may collapse when the information is extremely uneven. To avoid this issue, we also introduce a cross entropy function $\mathcal{L}_{sim}(\boldsymbol{p}_i, \tilde{\boldsymbol{q}}_{\boldsymbol{i}}) = (-\sum_{i=1}^{n} \tilde{\boldsymbol{q}}_{\boldsymbol{i}} \log \boldsymbol{p}_i)$ ($\boldsymbol{p}_i \in \boldsymbol{p}$) into the objective function. Integrating $\mathcal{L}_{sim}$ and $\mathcal{L}_c$ into the loss computation, we jointly train the classifier as well as the contrastive logical knowledge, and the ultimate loss function is given by:

$$\mathcal{L} = e \cdot \mathcal{L}_c + (1 - e) \cdot \mathcal{L}_{sim} \quad (e \in [0, 1)) \tag{8}$$

where e is a given parameter that control the degree of contrastive loss, larger e indicates that the model relies on contrastive learning more.

3.4 KNOWLEDGE REASONING

Based on the knowledge representation and contrastive logical knowledge learning, an interpretable structure $M = \langle S, R, R^*, \rho, s_0, \pi \rangle$ is learned from the CLK-based deep learning models. Specifically, for an arbitrary input sequence X_v, a classical TPK model has a form of tree where each word denotes the time step. With the logical knowledge embedd into models, explanations for sentiment analysis can be obtained based on the semantics of TPK. For example, as shown in Figure 1 (c) , the world under state s_3 is non-deterministic since there are different states on the same time step, and there is a public announcement at sate s'_3 such that $s'_3 \rho s_3$, so the model will change it's decision and go to another one. Based on such interpretable structures of TPK, we can provide explanations for users. Human-level knowledge dictates that explanations be logical, effective, and trustworthy;

causal reasoning can probe these qualities in depth. As a solution, we map a TPK tree to a structural causal model (SCM) (Shanmugam, 2001) to characterize the logic behind sentiment analysis tasks. Based on the learned TPK model, we decompose each sequence into the two variables: simple word I and public announcement T. Then we assume the following SCM:

$$c := f_C(U_C) \quad i := f_I(c, U_I) \quad t := f_T(c, U_T) \quad x := f_X(i, t, U_X) \quad y := f_Y(x, U_Y) \tag{9}$$

where C is the confounder that influences I and T; X is the input sequence influenced by I and T, Y is the predicted label of X and U^* represents the unmeasured variable. As shown in Figure 1 (d), SCM is represented intuitively by using a directed acyclic graph, where vertices are random variables, and directed edges represent the direct causality between these variables.Following the Lewis's notion of causality (Lewis, 1973; 1977; 1981), we verify the rationality and fidelity of explanations by generating counterfactual public announcements through a mathematical operator $do(t_0)$ interventions on learned public announcements. For instance, let ρ be a public announcement of s_1 at state s_n such that $s_n \rho s'_n$ and $s_1 R^* s_n$ and $s_1 R^* s'_n$; we can apply an intervention on variable T by fixing the value of T as t_0, denoted as $t := t_0$. Thus the intervention $t := t_0$ blocks the influence of the original T on X. The explanation is considered to be effective and trustworthy iff $\neg T \rightarrow \neg X$, since $\neg T \rightarrow \neg X$ means the SCM corresponding to the explanations is causally effective (Lewis, 1973; 1981).

4 EXPERIMENTS

4.1 BASELINES

We adopt both glass-box models and black-box models as our baselines, where the glass models includ Linear Regression(**LR**) (Molnar, 2022), K Nearest Neighbors (**KNN**) (Koech & Dombeu, 2021), Decision Tree (**DT**) (Zhu & Yang, 2016) and Explainable Boosting Machine (**EBM**) (Nori et al., 2019). Black-box baselines include **BLSTM**, **BGRU** and **Bert**, where **BLSTM** (Thireou & Reczko, 2007) and **BGRU** (Kim & Lee, 2017) are two widely used non-pretrained models for sentiment analysis; **Bert** is a pre-trained state-of-art model proposed in (Devlin et al., 2019), which has achieved high accuracy in many NLP tasks. For comparison, we also compare our method with other black-box explainability techniques such as Partial Dependence (Moosbauer et al., 2021), LIME (Ribeiro et al., 2016) and SHAP (Kokalj et al., 2021). Remark that, we understand that there are already many models achieving comparable performance combined with BLSTM (or BGRU), such as BLSTM-CNN (Shen et al., 2017) and BGRU-CNN (Wang et al., 2021). And there are also other variants of pre-trained models, such as Roberta (Tan et al., 2022). However, this paper aims to evaluate how interpretable methods influence base models (baselines), i.e., whether model performance is sacrificed when transparency is brought in. Therefore, we only employ the original deep learning models as our base models, even though our proposed method can combine with those variants to form interpretable ones by replacing the DNN model (in Figure 1 (a)) with them.

4.2 DATASETS AND EVALUATIONS

For model training and evaluation, we compiled four public sentiment analysis datasets including binary datasets and fine-grained datasets. Specifically, **IMDB** [2] and **Yelp-2** [3] are binary datasets with 2 label categories. **Weibo-8** [4] and **Yelp-4** are fine-grained datasets with more than 3 label categories. Note that **Yelp-2**, **Yelp-4** and **Weibo-8** are also unbalanced datasets and the datasets description can be found in Appendix C.1. As we mentioned in knowledge reasoning, it is important to measure not only how well a model performs but also how well it can be understood. For model performance, we employ the accuracy as evaluation metrics. For model interpretability, we employ evaluation metrics and causal reasoning to investigate it, where evaluation metrics are used to evaluate the degree of fit between the predicted labels by the model and the true labels in the datasets. Following (Dzikovska et al., 2012; 2013), we use macro average F_1-score as the primary evaluation metric since it is suitable for evaluating unbalanced class distributions (see Appendix C.1).

4.3 DETAILS OF IMPLEMENTATION

In our experiments, we set batch size as 128. For all input sequences, we set the maximum sequence lengths as 100 tokens. We set embedding size as 100 for non-pretrained models. For pre-trained

[2] https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[3] https://www.yelp.com/dataset

[4] http://tcci.ccf.org.cn/conference/2013/pages/page04_sam.html

models, we set the embedding size as 100 for English datasets and 300 for Chinese datasets. We used BERT-tiny Turc et al. (2019) for English datasets and ALBERT Lan et al. (2019) for Chinese datasets, which are both denoted as "Bert" in our experimental results. All models are trained with Adam Optimizer (Kingma & Ba, 2015) on GTX 2080 Ti and RTX 3090. We train our model's parameters with the default learning rate of adam in Keras. In our main experiments, we set the maximum epochs as 200, and we employ earlystopping to control model training with $min\_delta = 1e-5$ and $patience = 6$ (Graziotin & Abrahamsson, 2013). In our main experiments, parameter for contrastive logical knowledge learning are set as $e = 0.9$ and $margin = 0.1$ or $margin = 0.9$. We run 5 times and use the averaged performance with variance given in Appendix C.1.

5 Results and discussion

This part demonstrates how CLK performs on both binary sentiment analysis and fine-grained sentiment analysis, together with discusses the sensitivity of parameters and model interpretability.

5.1 Analysis on classification results

	IMDB				Yelp-2				Weibo-8				Yelp-4			
Models	**P**	**R**	**F_1**	**Acc**	**R**	**R**	**F_1**	**Acc**	**P**	**R**	**F_1**	**Acc**	**R**	**R**	**F_1**	**Acc**
LR$^{\diamond}$	52.00	51.98	51.92	52.00	44.19	50.00	46.92	88.39	10.37	12.52	9.71	60.91	44.19	50.00	46.92	88.39
KNN$^{\diamond}$	50.01	50.01	49.87	50.01	69.89	51.81	50.52	87.52	22.83	13.31	11.23	62.93	69.89	51.81	50.52	87.52
DT$^{\diamond}$	52.17	52.10	51.72	52.11	53.89	50.11	47.17	87.57	22.68	14.21	12.62	63.23	53.89	50.11	47.17	87.57
EBM$^{\diamond}$	61.85	61.84	61.83	61.84	87.81	51.71	50.18	88.12	12.03	12.61	9.91	62.88	87.81	51.71	50.18	88.12
BLSTM	81.96	81.79	81.77	81.79	79.75	76.97	77.25	90.14	37.39	16.08	15.95	42.53	72.89	70.47	71.41	71.06
BLSTM+LIME*	81.96	81.79	81.77	81.79	79.75	76.97	77.25	90.14	37.39	16.08	15.95	42.53	72.89	70.47	71.41	71.06
BLSTM+CLK	82.93	82.81	82.79	**82.81**	83.68	77.06	79.70$^{\wr}$	**91.98**	36.53	16.36	16.63$^{\wr}$	**45.28**	73.72	70.51	71.69$^{\wr}$	**71.41**
BGRU	82.73	82.57	82.54	82.57	84.63	74.82	78.48	91.86	47.13	16.40	17.76	56.56	71.19	71.04	71.00	70.99
BGRU+LIME*	82.73	82.57	82.54	82.57	84.63	74.82	78.48	91.86	47.13	16.40	17.76	56.56	71.19	71.04	71.00	70.99
BGRU+CLK	83.30	83.27	83.26	**83.27**	83.44	78.02	80.28$^{\wr}$	**92.05**	46.72	16.96	18.43$^{\wr}$	**58.02**	72.90	71.17	71.61$^{\wr}$	**71.48**
BERT	83.99	83.76	83.73	83.76	85.29	82.88	83.89	93.16	31.35	29.30	29.77	64.41	76.42	75.81	75.89	75.43
BERT+LIME*	83.99	83.76	83.73	83.76	85.29	82.88	83.89	93.16	31.35	29.30	29.77	64.41	76.42	75.81	75.89	75.43
BERT+CLK	84.21	84.13	84.12	**84.13**	86.53	83.26	84.74$^{\wr}$	**93.61**	36.89	32.84	33.59$^{\wr}$	**65.54**	77.34	76.44	76.77$^{\wr}$	**76.18**

Table 1: Results of non-pretrained models and Bert for sentiment analysis tasks, where $^{\diamond}$ denotes the glass model, LIME* denotes the interpretable techniques like LIME, such as SHAP and Partial Dependence (PD). $^{\wr}$ indicates the best macro F_1-score on unbalanced datasets and the bold number indicates the best Acc score over black-box baselines.

As shown in Table 1, our proposed CLK obtains quite similar or higher accuracy compared to the black-box baselines (BLSTM, BGRU and BERT), while offering a transparent view of the learning mechanism. When compared to glass-box models, CLK outperforms baselines in binary sentiment analysis tasks (IMDB and Yelp-2), while demonstrates a lower accuracy in fine-grained analysis tasks (Weibo-8 and Yelp-4). Due to the fact that Weibo-8 and Yelp-4 are fine-grained and unbalanced datasets, it is unreasonable to evaluate model performance solely based on accuracy. If readers pay close attention to the macro F_1 score, they may notice that CLK provides a significantly higher macro $F1$ score than all glass-box models. Given that all black-box baselines achieve lower accuracy and a significantly higher macro F_1 score in fine-grained analysis tasks than glass-box baselines, it is possible to conclude that glass-box models are more likely to classify samples as the class with high proportion in unbalanced datasets. Comparing with the explainability techniques like LIME, such as SHAP and Partial Dependence (denoted as LIME* in Table 1), we find that such methods are unable to enhance model performance even though they bring transparency to black-box models. In contrast, our method can boost model performance while improving model interpretability. It is also worth noting that although the precision (P) of CLK is lower than the some black-box baselines on Yelp-2 and Weibo-8, the recall (R), F_1-score and Acc results are always higher than baselines, which also demonstrates the advantages of CLK.

5.2 Sensitivity analysis of $margin$

To study the sensitivity of $margin$ in $\mathcal{L}_c$ (as stated in Equation 7), we apply our CLK in three deep learning models including BLSTM, BGRU and Bert. Figure 2 (a)-(f) shows the sensitivity

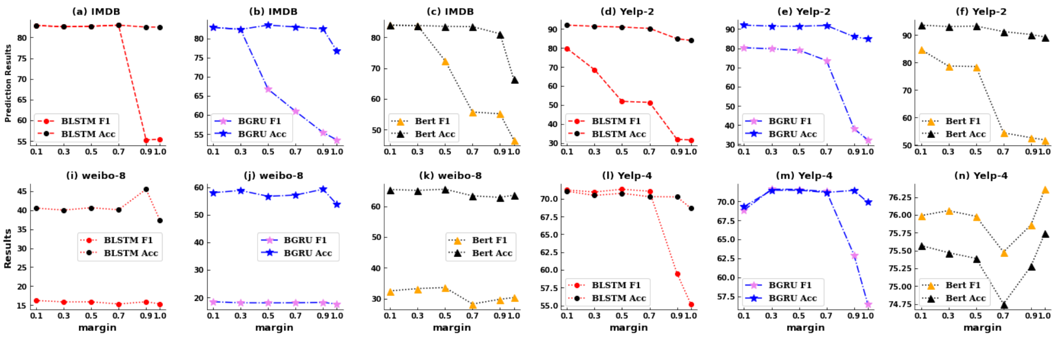


Figure 2: The sensitivity of $margin$ for sentiment analysis with $e = 0.9$, where the horizontal axis represents the value of $margin$, and the vertical axis represents the prediction results.

of $margin$ in binary sentiment analysis datasets. When $margin > 0.5$, we observed that the performance of most CLK-based models rapidly declined, with the F_1 score dropping more quickly than the Acc. Figure 2 (i)-(n) illustrates the sensitivity of $margin$ in fine-grained sentiment analysis datasets. The performance of CLK based models slightly fluctuate on Weibo-8 with F_1 score much lower that accuracy. As for Yelp-4 dataset, the performance of CLK based BLSTM (or BGRU) rapidly degrades when $margin > 0.7$, while CLK-based Bert varies with the value of $margin$. Note that macro F_1-scores of all CLK based models are lower than Acc except for CLK-Based BLSTM on Yelp-4. Besides, despite the significance of contrastive learning, it should be weighted with a proper value for good performance.

5.3 Ablation Test

We conduct an ablation test on Weibo-8 dataset since Weibo-8 is a fine-grained unbalanced sentiment analysis datatset, we mainly focus on the macro F_1-score. During training stage, our method (CLK) mainly includes knowledge acquisition (denoted as KA) and knowledge learning (KL). We demonstrate the effect of each component. As shown in Table 2, BLSTM benefits a lot from adding knowledge extraction (KA), whereas BGRU and BERT suffer when equipped with KA but not knowledge learning (KL). And the performance of all baselines drops when only equipped with the initial label embedding (IE) method. This phenomenon is triggered by the properties of the data; Weibo-8 is a fine-grained dataset with some indistinguishable labels, such as happiness and like, disgust and sadness. In addition, Weibo-8 is also an unbalanced dataset, making it more challenging to classify sequences into indistinguishable labels.

Models	**P**	**R**	$\boldsymbol{F_1}$	**Acc**
BLSTM	37.39	16.08	15.95	42.53
+IE	34.55	15.60	15.69	43.77
+KA	40.59	16.49	$16.99^{\wr}$	**47.37**
+IE+KL	32.43	15.06	13.63	35.26
+CLK	36.53	16.36	16.63	45.28
BGRU	47.13	16.40	17.76	56.56
+IE	39.37	16.30	15.75	37.12
+KA	36.65	16.51	16.59	41.55
+IE+KL	48.48	16.99	18.29	57.60
+CLK	46.72	16.96	$18.43^{\wr}$	**58.02**
BERT	31.35	29.30	29.77	64.41
+IE	7.70	12.22	9.45	63.01
+KA	7.70	12.22	9.45	63.02
+IE+KL	33.87	32.15	32.44	63.69
+CLK	36.89	32.84	$33.59^{\wr}$	**65.54**

Table 2: Abalation test on Weibo-8, where "+KA" indicates baselines equipped with knowledge acquisition, "+KL" indicates baselines equipped with knowledge learning, "+IE" indicates baselines equipped with random initial label embedding and "+CLK" means "+KA+KL".

Readers paying close attention to Table 2 may notice that IE and KA reduce the performances of BERT to the same level, this is mainly due to the anisotropy and uncertainty of logical knowledge as we mentioned in Section 1, where the anisotropy is mainly produced when using pre-trained models (Cai et al., 2020; Ethayarajh, 2019). What's more, when equipped with knowledge learning, BGRU and BERT can be improved by a large margin, which indicates that ineffective knowledge not only does not improve the model, but also rather hinders its performance by clouding its decision-making. Hence, contrastive knowledge learning is essential and effective when learning high-quality representations for indistinguishable labels with unbalanced distribution.

5.4 A Case Study for Interpretability Through Knowledge Reasoning

Interpretability analysis makes it possible for us to quickly and simply understand how the model arrives at its predictions. In this case study, we demonstrate the interpretable result from a trained model (Bert+CLK) with a real positive example in Yelp-2 test data.

Example 1 *Consider the sequence $task_{1001}$ =“We went for lunch. The waitress took the bread basket from another table with dirty dishes on it and put it on our table. Open Grated cheese dish goes around from table to table . This is not sanitary.”*

Question: Why the corresponding prediction is “negative” rather than “positive”?

Explanation: Because the “not sanitary” (public announcement) appears in the following words, so the model classifies this sequence into “negative”.

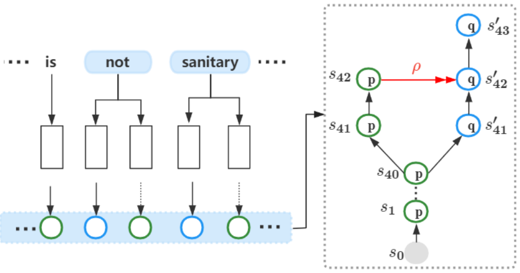


Figure 3: The TPK model of **Example 1**

As shown in Figure 3, we can visualize the explanation of this example in the form of TPK based on the knowledge representation in Section 3. In the diagram of TPK models, the arrow $\cdot\!\!\rightarrow$ represents successor relation R, double arrow $\twoheadrightarrow$ represents the relation ρ, and s_0 is the root of the tree structure. Let the proposition letter p denote that the sequence is recognized as “positive”, proposition letter q denote that the sequence is recognized as “negative”.

Then $task_{1003}$ is correctly classified iff $(s_{43}' \vDash q)$, where s_i and s_i' denote the states of i^{th} word in the sequence. The interpretable results are shown in Figure 3, where $s_{40} \vDash Yp$. At state s_{42}, we recognise that the reason path with $\{s_0, s_1, s_3, \ldots, s_{42}\}$ contains an incompatibility because it uses the choice of “positive” and this is incompatible with “not sanitary”. Thus, there is a public announcement that $s_{42}\rho s_{42}'$, so we have $s_{42} \vDash \boxminus q$ and $s_{42}' \vDash \mathbb{Y}p$. Then, the reason path becomes $\{s_0, s_1, \ldots, s_{42}, s_{42}', \ldots, s_{43}'\}$ and ultimately the $task_{1003}$ is classified as “negative” with $s_{43}' \vDash q$. From the perspective of human knowledge, it is clear that $task_{1003}$ is correctly classified, but it’s still unknown that whether the explanation based on the TPK model in Figure 3 is trustworthy.

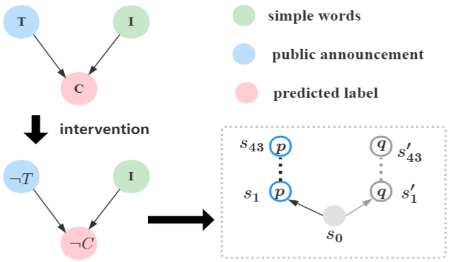


Figure 4: The counterfactual TPK model.

As mentioned in subsection 3.4, we investigate the fidelity of the explanation by counterfactual reasoning. Here we generates a counterfactual public announcement for ρ through a mathematical operator $do(t_0)$ interventions on $task_{1003}$. Then we can obtain a counterfactual TPK model as shown in Figure 4, where $s_{43} \vDash Yp$ and $s_1 \vDash \Box p$, and the corresponding reason path is $\{s_0, s_1, \ldots, s_{43}\}$. In this counterfactual case, $task_{1003}$ is classified as “negative”, which indicates that the public announcement ρ is causally effective since $\neg T \rightarrow \neg C$. Therefore, we can conclude that the model derives the prediction “negative” because the “not sanitary” (public announcement) appears; and the model would classify $task_{1003}$ as “positive” without the public announcement. Based on above analysis, it’s possible to derive some intuitions from the point of model interpretability. In contrast to existing interpretable methods like LIME, SHAP, and PD, our method generates explanations by learning interpretable structures (TPK) from deep learning models. Besides, our method is proven to be causally successful by counterfactual reasoning, which is a key distinction between CLK and other interpretable methods like LIME.

6 Conclusion

In this paper, we present a novel contrastive logical knowledge learning (CLK) method to learn interpretable TPK models and generate explanations for sentiment analysis tasks. To the best of our knowledge, this is the first work that uses contrastive learning to optimize the learning of logical knowledge from datasets and trained models. Our method achieves comparable performance while adding transparency and interpretability to deep learning models by leveraging the benefits of both logical knowledge and contrastive learning. Empirically, we conducted extensive experiments to show the effectiveness of CLK, and our results are competitive both in binary sentiment analysis tasks and fine-grained sentiment analysis tasks. The case study for model interpretability shows that CLK can generate human-understandable explanations, and the knowledge reasoning of explanations demonstrates the rationality and fidelity of our method. Our research aims to raise awareness of the potential of modal logic and contrastive learning in delivering interpretable sentiment analysis.

REFERENCES

Michael Abraham, Israel Belfer, Dov M. Gabbay, and Uri J. Schild. Future determination of entities in talmudic public announcement logic. *J. Appl. Log.*, 11(1):63–90, 2013. doi: 10.1016/j.jal.2012.06.001. URL https://doi.org/10.1016/j.jal.2012.06.001.

Thomas Ågotnes and Natasha Alechina. Semantics for dynamic syntactic epistemic logics. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty (eds.), *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, pp. 411–419. AAAI Press, 2006. URL http://www.aaai.org/Library/KR/2006/kr06-043.php.

Shivaji Alaparthi and Manit Mishra. Bidirectional encoder representations from transformers (BERT): A sentiment analysis odyssey. *CoRR*, abs/2007.01127, 2020. URL https://arxiv.org/abs/2007.01127.

Fernando X. Arias, Mayteé Zambrano Núñez, Ariel Guerra-Adames, Nathalia Tejedor-Flores, and Miguel Vargas-Lombardo. Sentiment analysis of public social media as a tool for health-related topics. *IEEE Access*, 10:74850–74872, 2022. URL https://doi.org/10.1109/ACCESS.2022.3187406.

Pietro Barbiero, Gabriele Ciravegna, Francesco Giannini, Pietro Lió, Marco Gori, and Stefano Melacci. Entropy-based logic explanations of neural networks. In *Thirty-Sixth AAAI Conference on Artificial Intelligence,*, pp. 6046–6054. AAAI Press, 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/20551.

Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, and U. Rajendra Acharya. ABCDM: an attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Gener. Comput. Syst.*, 115:279–294, 2021. URL https://doi.org/10.1016/j.future.2020.08.005.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003. URL http://jmlr.org/papers/v3/blei03a.html.

Rbra Bull. An introduction to modal logicby e. j. lemmon; dana scott; krister segerberg. *The Journal of Symbolic Logic*, 44(4):653–654, 1979.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. Isotropy in the contextual embedding space: Clusters and manifolds. In *International Conference on Learning Representations*, 2020.

Yulin Chen, Zelai Yao, Haixiao Chi, Dov M. Gabbay, Bo Yuan, Bruno Bentzen, and Beishui Liao. Btpk-based learning: An interpretable method for named entity recognition. *CoRR*, abs/2201.09523, 2022. URL https://arxiv.org/abs/2201.09523.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James R. Glass. Diffcse: Difference-based contrastive learning for sentence embeddings. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pp. 4207–4218. Association for Computational Linguistics, 2022. URL https://doi.org/10.18653/v1/2022.naacl-main.311.

Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic. Equivariant contrastive learning. *CoRR*, abs/2111.00899, 2021. URL https://arxiv.org/abs/2111.00899.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/n19-1423.

Myroslava O. Dzikovska, Rodney D. Nielsen, and Chris Brew. Towards effective tutorial feedback for explanation questions: A dataset and baselines. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, Montréal, Canada*, pp. 200–210. The Association for Computational Linguistics, 2012. URL https://aclanthology.org/N12-1021/.

Myroslava O. Dzikovska, Elaine Farrow, and Johanna D. Moore. Combining semantic interpretation and statistical classification for improved explanation processing in a tutorial dialogue system. In H. Chad Lane, Kalina Yacef, Jack Mostow, and Philip I. Pavlik (eds.), *Artificial Intelligence in Education - 16th International Conference, AIED, Memphis, TN, USA, July 9-13*, volume 7926 of *Lecture Notes in Computer Science*, pp. 279–288. Springer, 2013. URL https://doi.org/10.1007/978-3-642-39112-5_29.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings. *CoRR*, abs/1909.00512, 2019. URL http://arxiv.org/abs/1909.00512.

Yixuan Fan, Zhaopeng Dou, Yali Li, and Shengjin Wang. Portrait interpretation and a benchmark. *CoRR*, abs/2207.13315, 2022. doi: 10.48550/arXiv.2207.13315. URL https://doi.org/10.48550/arXiv.2207.13315.

João Ferreira, Manuel de Sousa Ribeiro, Ricardo Gonçalves, and João Leite. Looking inside the black-box: Logic-based explanations for neural networks. In Gabriele Kern-Isberner, Gerhard Lakemeyer, and Thomas Meyer (eds.), *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022, Haifa, Israel. July 31 - August 5, 2022*, 2022. URL https://proceedings.kr.org/2022/45/.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 6894–6910. Association for Computational Linguistics, 2021. URL https://doi.org/10.18653/v1/2021.emnlp-main.552.

John M. Giorgi, Osvald Nitski, Bo Wang, and Gary D. Bader. Declutr: Deep contrastive learning for unsupervised textual representations. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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), Virtual Event, August 1-6, 2021*, pp. 879–895. Association for Computational Linguistics, 2021. URL https://doi.org/10.18653/v1/2021.acl-long.72.

Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *CoRR*, abs/2008.05756, 2020. URL https://arxiv.org/abs/2008.05756.

Daniel Graziotin and Pekka Abrahamsson. A web-based modeling tool for the SEMAT essence theory of software engineering. *CoRR*, abs/1307.2075, 2013. URL http://arxiv.org/abs/1307.2075.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pp. 1735–1742. IEEE Computer Society, 2006. URL https://doi.org/10.1109/CVPR.2006.100.

Keijo Heljanko et al. *Model checking the branching time temporal logic CTL.* Citeseer, 1997.

Tomoki Ito, Kota Tsubouchi, Hiroki Sakaji, Tatsuo Yamashita, and Kiyoshi Izumi. Word-level contextual sentiment analysis with interpretability. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 4231–4238. AAAI Press, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/5845.

Rbd Kaplan. Semantical analysis of modal logic i. normal modal propositional calculiby saul a. kripke. *Journal of Symbolic Logic*, 31(1):120–122, 1966.

Jonggu Kim and Jong-Hyeok Lee. Multiple range-restricted bidirectional gated recurrent units with attention for relation classification. *CoRR*, abs/1707.01265, 2017. URL http://arxiv.org/abs/1707.01265.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Gideon Koech and Jean Vincent Fonou Dombeu. K-nearest neighbors classification of semantic web ontologies. In J. Christian Attiogbé and Sadok Ben Yahia (eds.), *Model and Data Engineering - 10th International Conference, MEDI 2021, Tallinn, Estonia, June 21-23, 2021, Proceedings*, volume 12732 of *Lecture Notes in Computer Science*, pp. 241–248. Springer, 2021. URL https://doi.org/10.1007/978-3-030-78428-7_19.

Enja Kokalj, Blaz Skrlj, Nada Lavrac, Senja Pollak, and Marko Robnik-Sikonja. BERT meets shapley: Extending SHAP explanations to transformer-based classifiers. In Hannu Toivonen and Michele Boggia (eds.), *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation, EACL 2021, Online, April 19, 2021*, pp. 16–21. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.hackashop-1.3/.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019. URL http://arxiv.org/abs/1909.11942.

David Lewis. Counterfactuals and comparative possibility. *J. Philos. Log.*, 2(4):418–446, 1973. doi: 10.1007/BF00262950. URL https://doi.org/10.1007/BF00262950.

David Lewis. Possible-world semantics for counterfactual logics: A rejoinder. *J. Philos. Log.*, 6(1): 359–363, 1977. doi: 10.1007/BF00262070. URL https://doi.org/10.1007/BF00262070.

David Lewis. Ordering semantics and premise semantics for counterfactuals. *J. Philos. Log.*, 10(2): 217–234, 1981. doi: 10.1007/BF00248850. URL https://doi.org/10.1007/BF00248850.

Bryan Li, Dimitrios Dimitriadis, and Andreas Stolcke. Acoustic and lexical sentiment analysis for customer service calls. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pp. 5876–5880. IEEE, 2019. doi: 10.1109/ICASSP.2019.8683679. URL https://doi.org/10.1109/ICASSP.2019.8683679.

Han Liu and Mihaela Cocea. Fuzzy rule based systems for interpretable sentiment analysis. In *Ninth International Conference on Advanced Computational Intelligence, ICACI 2017, Doha, Qatar, February 4-6, 2017*, pp. 129–136. IEEE, 2017. doi: 10.1109/ICACI.2017.7974497. URL https://doi.org/10.1109/ICACI.2017.7974497.

Xinghan Liu and Emiliano Lorini. A logic for binary classifiers and their explanation. In Pietro Baroni, Christoph Benzmüller, and Yì N. Wáng (eds.), *Logic and Argumentation - 4th International Conference, CLAR 2021, Hangzhou, China, October 20-22, 2021, Proceedings*, volume 13040 of *Lecture Notes in Computer Science*, pp. 302–321. Springer, 2021. URL https://doi.org/10.1007/978-3-030-89391-0_17.

Ling Luo, Xiang Ao, Feiyang Pan, Jin Wang, Tong Zhao, Ningzi Yu, and Qing He. Beyond polarity: Interpretable financial sentiment analysis with hierarchical query-driven attention. In Jérôme Lang (ed.), *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 4244–4250. ijcai.org, 2018. URL https://doi.org/10.24963/ijcai.2018/590.

Christoph Molnar. *Model-agnostic interpretable machine learning*. PhD thesis, Ludwig Maximilian University of Munich, Germany, 2022. URL https://edoc.ub.uni-muenchen.de/30374/.

Julia Moosbauer, Julia Herbinger, Giuseppe Casalicchio, Marius Lindauer, and Bernd Bischl. Explaining hyperparameter optimization via partial dependence plots. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman

Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 2280–2291, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/12ced2db6f0193dda91ba86224ea1cd8-Abstract.html.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. Fine-grained sentiment classification using BERT. *CoRR*, abs/1910.03474, 2019. URL http://arxiv.org/abs/1910.03474.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability. *CoRR*, abs/1909.09223, 2019. URL http://arxiv.org/abs/1909.09223.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1532–1543. ACL, 2014. doi: 10.3115/v1/d14-1162. URL https://doi.org/10.3115/v1/d14-1162.

Isidoros Perikos, Spyridon Kardakis, and Ioannis Hatzilygeroudis. Sentiment analysis using novel and interpretable architectures of hidden markov models. *Knowl. Based Syst.*, 229:107332, 2021. doi: 10.1016/j.knosys.2021.107332. URL https://doi.org/10.1016/j.knosys.2021.107332.

Kumar Ravi and Vadlamani Ravi. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowl. Based Syst.*, 89:14–46, 2015. URL https://doi.org/10.1016/j.knosys.2015.06.015.

Nils Reimers and Iryna Gurevych. Alternative weighting schemes for elmo embeddings. *CoRR*, abs/1904.02954, 2019. URL http://arxiv.org/abs/1904.02954.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (eds.), *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144. ACM, 2016. URL https://doi.org/10.1145/2939672.2939778.

Nuttapong Sanglerdsinlapachai, Anon Plangprasopchok, Tu Bao Ho, and Ekawit Nantajeewarawat. Improving sentiment analysis on clinical narratives by exploiting UMLS semantic types. *Artif. Intell. Medicine*, 113:102033, 2021. doi: 10.1016/j.artmed.2021.102033. URL https://doi.org/10.1016/j.artmed.2021.102033.

Ram Shanmugam. Causality: Models, reasoning, and inference : Judea pearl; cambridge university press, cambridge, uk, 2000, pp 384, ISBN 0-521-77362-8. *Neurocomputing*, 41(1-4):189–190, 2001. doi: 10.1016/S0925-2312(01)00330-7. URL https://doi.org/10.1016/S0925-2312(01)00330-7.

Qianzi Shen, Zijian Wang, and Yaoru Sun. Sentiment analysis of movie reviews based on CNN-BLSTM. In *Intelligence Science I - Second IFIP TC 12 International Conference, ICIS 2017, Shanghai, China, October 25-28, 2017, Proceedings*, volume 510 of *IFIP Advances in Information and Communication Technology*, pp. 164–171. Springer, 2017. doi: 10.1007/978-3-319-68121-4_17. URL https://doi.org/10.1007/978-3-319-68121-4_17.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pp. 1631–1642. ACL, 2013. URL https://aclanthology.org/D13-1170/.

Kian Long Tan, Chin-Poo Lee, Kalaiarasi Sonai Muthu Anbananthen, and Kian-Ming Lim. Roberta-lstm: A hybrid model for sentiment analysis with transformer and recurrent neural network. *IEEE Access*, 10:21517–21525, 2022. doi: 10.1109/ACCESS.2022.3152828. URL https://doi.org/10.1109/ACCESS.2022.3152828.

Trias Thireou and Martin Reczko. Bidirectional long short-term memory networks for predicting the subcellular localization of eukaryotic proteins. *IEEE ACM Trans. Comput. Biol. Bioinform.*, 4(3):441–446, 2007. doi: 10.1145/1299023.1299033. URL http://doi.acm.org/10.1145/1299023.1299033.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962, 2019. URL http://arxiv.org/abs/1908.08962.

David Valle-Cruz, Vanessa Fernandez-Cortez, Asdrúbal López Chau, and Rodrigo Sandoval-Almazán. Does twitter affect stock market decisions? financial sentiment analysis during pandemics: A comparative study of the H1N1 and the COVID-19 periods. *Cogn. Comput.*, 14(1):372–387, 2022. doi: 10.1007/s12559-021-09819-8. URL https://doi.org/10.1007/s12559-021-09819-8.

Qingqing Wang, Jianglin Luo, and Jianwen Song. Emotion analysis method for elderly living alone based on CNN-BGRU neural network. *Int. J. Wirel. Mob. Comput.*, 20(4):352–362, 2021. doi: 10.1504/IJWMC.2021.117557. URL https://doi.org/10.1504/IJWMC.2021.117557.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9929–9939. PMLR, 2020. URL http://proceedings.mlr.press/v119/wang20k.html.

Xiaoxu Wang, Chaofeng Li, Jun Zhang, Qianyun Zhang, and Jinwen Hu. Variational inference -based EM for quantized FIR system parameter identification. In *14th IEEE International Conference on Control and Automation, ICCA 2018, Anchorage, AK, USA, June 12-15, 2018*, pp. 636–640. IEEE, 2018. doi: 10.1109/ICCA.2018.8444211. URL https://doi.org/10.1109/ICCA.2018.8444211.

Jiyao Wei, Jian Liao, Zhenfei Yang, Suge Wang, and Qiang Zhao. Bilstm with multi-polarity orthogonal attention for implicit sentiment analysis. *Neurocomputing*, 383:165–173, 2020. URL https://doi.org/10.1016/j.neucom.2019.11.054.

Rohan Kumar Yadav, Lei Jiao, Ole-Christoffer Granmo, and Morten Goodwin. Human-level interpretable learning for aspect-based sentiment analysis. 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, Virtual Event, February 2-9, 2021*, pp. 14203–14212. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17671.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.), *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 1480–1489. The Association for Computational Linguistics, 2016. URL https://doi.org/10.18653/v1/n16-1174.

Qi Zhang, Jin Qian, Huan Chen, Jihua Kang, and Xuan-Jing Huang. Discourse level explanatory relation extraction from product reviews using first-order logic. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 946–957, 2013.

Lin Zhu and Yang Yang. Improvement of decision tree ID3 algorithm. In Shangguang Wang and Ao Zhou (eds.), *Collaborate Computing: Networking, Applications and Worksharing - 12th International Conference, CollaborateCom 2016, Beijing, China, November 10-11, 2016, Proceedings*, volume 201 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 595–600. Springer, 2016. URL https://doi.org/10.1007/978-3-319-59288-6_59.

Slavko Zitnik, Neli Blagus, and Marko Bajec. Target-level sentiment analysis for news articles. *Knowl. Based Syst.*, 249:108939, 2022. doi: 10.1016/j.knosys.2022.108939. URL https://doi.org/10.1016/j.knosys.2022.108939.

A RELATED WORK

Our work adds transparency to deep learning models by learning interpretable TPK models and generating explanations for sentiment analysis tasks. Additionally, our work extends the logic-based interpretable method and provides insight into the inner working of deep learning models beyond visualisation and is adaptable to a variety of explanatory settings. In this section, we briefly review interpretable approaches for sentiment analysis tasks, contrastive learning and logic-based methods for providing explanations.

Interpretable sentiment analysis There have been numerous studies on sentiment analysis. Early sentiment analysis mainly uses lexicon-based methods for text classification, which mainly depend on dictionaries and corpora (Ravi & Ravi, 2015). Because of the language's adaptability and lack of formality, it is difficult to construct a general and efficient rule applicable to all contexts. Many works utilize machine learning and deep learning models to accomplish sentiment analysis, for example, Basiri et al. (2021) uses a bidirectional neural network architecture for sentiment analysis, Wei et al. (2020) uses a BLSTM with multi-polarity orthogonal attention, and Yang et al. (2016) uses a hierarchical attention network (HAN). In recent years, with the development of pre-trained language model, Reimers & Gurevych (2019) utilizes the ELMo embeddings and Munikar et al. (2019) utilizes BERT deep learning model to improve the performance of sentiment analysis tasks. Although these models can effectively capture the semantic relationships between words in context, they lack interpretability for sentiment analysis. Consequently, several studies have been conducted to investigate the interpretability of models for sentiment analysis. Liu & Cocea (2017) proposes the fuzzy rule based systems, which can reduce the computational complexity and increase the interpretability. Luo et al. (2018) proposes a query-driven attention mechanism to discover different spotlights for queries from different aspects and provide explainable results. Perikos et al. (2021) introduces an efficient method based HMM for sentiment analysis, and Yadav et al. (2021) provides an interpretable learning approach for the aspect-based sentiment analysis tasks. Despite their successes, none of the existing work attempted to leverage deep learning models (particularly pre-trained language models) with human-readable explanations for sentiment analysis tasks, achieving the state-of-the-art performance.

Contrastive learning. Contrastive learning is a learning paradigm which aims to teach models to distinguish the targed sample from its corresponing negative samples. This method has been extensively used in numerous research fields. In the field of natural language process, numerous contrastive approaches have been utilized on sentence-level. For example, Gao et al. (2021) presents SimCSE, a simple contrastive learning framework that greatly enhances state-of-the-art sentence embeddings on semantic textual similarity tasks. Giorgi et al. (2021) proposes a simple framework, DeCLUTR, to shorten the distance between sentence embeddings in the same text and to widen the distance between sentence embeddings in different text. Inspired by equivariant contrastive learning (Dangovski et al., 2021), the generalization of contrastive learning in computer vision, Chuang et al. (2022) propoeses DiffCSE to learn the sentence embeddings between the original sentence and the edited sentence. Unlike these previous works, this paper utilizes contrastive learning for the first time (to the best of our knowledge) to enhance the uniformity and reduce the uncertainty of the logical knowledge so as to make logical explanation consistent with real knowledge.

Logic-based interpretable methods. There have been plenty of studies employing logic-based methods to provide explanations in explainable artificial intelligence domains. For example, Ferreira et al. (2022) develops a procedure to induce logic-based explanations for a given neural network model. Liu & Lorini (2021) presents a modal language of a ceteris paribus nature which can provide explanations for binary input classifiers. Barbiero et al. (2022) proposes an end-to-end differentiable approach which can obtain logic explanations from neural networks using the formalism of First-Order Logic. Although these work, the mechanism of deep learning models is still unknown. To address this issue, (Chen et al., 2022) first investigates the working mechanism of Gate BRNNs through Talmudic Public Announcement logic. Enlightened by it, this paper employs Talmudic Public Announcement logic to generate explanations for sentiment analysis, and further applies proposed method to pre-trained models (Bert) to get interpretable ones, which is missing from existing works.

B The detailed background of Talmudic public announcement logic

This part provides a general introduction of the kripke model in Modal logic as a preliminary knowledge, and then introduce the basic knowledge of TPK.

B.1 Kripke model

Modal logic (Bull, 1979) is typically used to describe the systems which have the expression related to "necessarily" or "possibly". And it may be broadly used in belief, deontic, temporal and other systems. Modal logic $\mathbf{K}$ is formulated in the language of classical logic with the added unary operator $\Box$. There are a range of different types of modal logics, all governed by a similar set of logical axioms and rules. Kripke (1963) (Kaplan, 1966) defined semantics which can be applied over all the modal logics. Here, we introduce the Kripke model for time action logic.

A Kripke model (or Kripke structure) M for time action logic is a tuple (S, R, Ω, h), where (S, R, Ω) is a tree with root Ω,

- S: A set of possible worlds.
- R: The accessibility binary relation among the worlds, i.e. $R \subseteq S \times S$. We write $s_1 R s_2$ to indicate that $(s_1, s_2) \in R$;
- h: An assignment giving for each atom q of the language a subset $h(q) \in S$.

Satisfaction is of the form $t \vDash A$, where $t \in S$ and A a wff and is defined recursively as follows:

1. $t \vDash q$ iff $t \in h(q)$ for q atomic.
2. $t \vDash A \wedge B$ iff $t \vDash A$ and $t \vDash B$.
3. $t \vDash \neg A$ iff $t \nvDash A$.
4. $t \vDash \Box A$ iff for all s such that s is an immediate successor of t in the tree we have $s \models A$.
5. A holds in the model (S, R, Ω, h) iff $\Omega \models A$.

So the Kripke model of time action logic could always be depicted as a tree structure using successor functions, which is called a time action model $\mathbf{m} = (\mathbb{R}, R, \Omega)$,

- $\mathbb{R} = \mathbf{r_1}, \mathbf{r_2}, \ldots$ is a set of unary successor functions capable of opereting on Ω. Thus $t = \Omega \mathbf{a_1 a_2 a_3} \ldots \mathbf{a_n}$ is the form of element sequence. The elements of $\mathbb{R}$ could be regarded as actions $\mathbf{a} \in \mathbb{R}$ moving the agent from any state t to a new state $t\mathbf{a}$.
- R is the accessibility binary relation among the states. tRs is hold iff for some $\mathbf{a} \in \mathbb{R}$ we have $s = t\mathbf{a}$
- Ω is the initial state.

Time comes into the model if we take the view that time moves one unit when the agent perform any action. So the states can also represent moments, i.e. Time 0, Time 1 . . . and the paths represent how time moves. There is always one root in the graph, which means the initial state, time 0.

The ordinary dynamic logics deal with actions upon states, whose possible results are clear cut. However, in some certain cases, actions may depend on the future and therefore may be not clear cut at the present and need future clarifications. Thus, a public announcement mechanism has been put forward in (Abraham et al., 2013).

B.1.1 Public announcement

To deal with the future clarifications, a set of public announcements $\mathbb{P}$ is defined in a Talmudic $\mathbf{K}$ frame(Abraham et al., 2013). A Talmudic $\mathbf{K}$ frame has the form $(S, \mathcal{R}, \mathbb{P})$ where

- S is a non-empty set of possible worlds.

- $\mathcal{R}$ is a multi-valued accessibility relation of the form xRY_x, where $Y_x = x_1, \ldots, x_{n(x)}$, which means: one of x_i in Y_x is accessible to x but it is not clear which one and need to await a public announcement clarification. The set $Y_x = x_1, \ldots, x_{n(x)}$ is unique in its $\mathcal{R}$ relation to x.

  $(S, \mathcal{R})$ is a tree if for any y in S there is at most one x in S such that y is in Y_x and $x\mathcal{R}Y_x$ holds. And define $n(x)$, for x in S the accessibility branching parameter of x.
- $\mathbb{P}$ is a set of public announcements with the deterministic form: $\alpha = (x, x_1, ..., x_{n(x)}, y)$, $y \in x_1, ..., x_{n(x)}$ where $(x, x_1, ..., x_{n(x)}, y) \in \mathcal{R}$ means: it is hereby announced that y is the element accessible to x.

  While the public announcement can be non-deterministic if it chooses a subset Y of $x_1, ..., x_{n(x)}$. It therefore has the form $\alpha = (x, x_1, ..., x_{n(x)}, Y)$, Y is a subset of $x_1, ..., x_{n(x)}$. If Y is allowed to be empty, this means that it is announced that x has no accessible points.

The public announcement in Talmudic frame is a revision on the relation $\mathcal{R}$, say, we have $(S, \mathcal{R}, t)$ and announce α, then we move to $(S, \mathcal{R}_\alpha, t)$. A certain α_1 not only expresses the public announce, but also tells where the current state is supposed to go to, and also give the information about what time and place the public announcement is made.

B.2 Definition of a TPK model

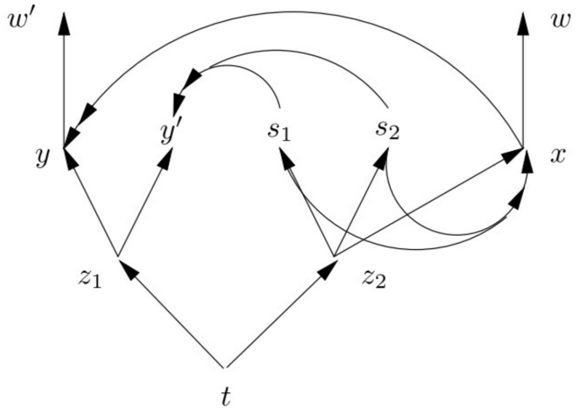


Figure 5: A non-deterministic TPK model (Abraham et al., 2013)

Based on above knowledge, a Talmudic public announcement logic model is defined as shown in Section 2. Remark that, a deterministic TPK model is assumed to be deterministic because the public announcement ρ leads to at most one successor state when there is a public announcement, i.e., one state should be the unique clarified successor of the state where public announcement occurs. To make it clear, we now give an example of deterministic TPK and non-deterministic TPK, respectively.

An example of non-deterministic TPK model is shown in shown in Figure 5. Here the arrow → represents successor relation R, double arrow ↠ represents the relation ρ, and s_0 is the root of the tree structure.

- It is not a deterministic model, because $(s_1 \rho y' \wedge s_1 \rho x \wedge y' \neq x)$ holds. (or the same for s_2)
- Some equivalence relation of distance could be raised, like $D(y) = D(x)$, since we have $x\rho y \wedge \neg(yRx) \wedge y \neq x$.
- t is publicly clarified at z_2 to z_1, expressed by $s_1 \twoheadrightarrow y'$ or $s_2 \twoheadrightarrow y'$ or $x \twoheadrightarrow y$, since we have $s_1\rho y'$, $s_2\rho y'$, $x\rho y$, with z_1Ry', z_1Ry.
- The double arrow ↠ also shows that where the announcement points are supposed to go to, i.e. s_2 or s_2 are supposed to go to y' when the public announcement at z_2 is made.

An example of deterministic TPK model is shown in shown in Figure 6. Similarly, the arrow $\rightarrow$ represents successor relation R, double arrow $\twoheadrightarrow$ represents the relation ρ, and s_0 is the root of the tree structure.

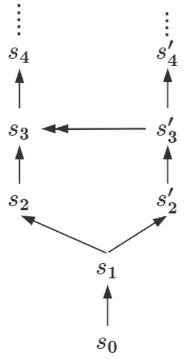


Figure 6: A deterministic TPK model

- It is a deterministic TPK model, because the public announcement ρ is deterministic and leads to at most one successor state, i.e. $s'_3 \rho s_3$.
- Some equivalence relation of distance could be raised, like $D(s_3) = D(s'_3)$, since we have $s'_3 \rho s_3 \wedge \neg(s'_3 R s_3) \wedge s_3 \neq s'_3$.
- The double arrow $\twoheadrightarrow$ also shows that where the announcement points are supposed to go to, i.e. s_1 is supposed to go to s_2 when the public announcement at s'_3 is made.

C THE SUPPLEMENT OF EXPERIMENTS

C.1 DATASET DESCRIBTION AND EVALUATION

For model training and evaluation, we compiled four public sentiment analysis datasets including binary datasets and fine-grained datasets.

- **IMDB** [5] is a common used dataset for binary sentiment analysis containing 5,0000 available sequences, where “positive:negative=1:1”.
- **Yelp-2** [6] is a review corpus obtained from Yelp website. We process the raw data and extract 10937 available sequences with 2 label categories, where “positive:negative=9573:1364”.
- **Weibo-8** [7] is a fine-grained dataset provided by CCF for fine-grained sentiment analysis. We extract 13250 available sequences with 8 label categories, where “none:like:disgust:sadness:happiness:anger:surprise:fear= 8313:1224:1004:838:728:718:310:115”.
- **Yelp-4** [8] is a review corpus obtained from Yelp website. We process the raw data and extract 10937 available sequences with 4 label categories, where “positive:slightly positive:slightly negative:negative= 2391:4132:3050:1364”.

The division of training samples, validation samples, and test samples is shown in Table 3.

[5] https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[6] https://www.yelp.com/dataset

[7] http://tcci.ccf.org.cn/conference/2013/pages/page04_sam.html

[8] https://www.yelp.com/dataset

dataset	sequences	train samples	validation samples	test samples	label cataegories
IMDB	50000	250000	5000	2500	2
Yelp-2	10937	7437	1859	1641	2
Weibo-8	13250	9010	2252	1988	8
Yelp-4	10937	7875	1968	1094	4

Table 3: Datasets description and partition.

Macro average F_1-score As shown in Table 3, Yelp-2, Yelp-4, and Weibo-8 are imbalanced datasets, especially Weibo-8. Therefore, appropriate metrics should be used to evaluate the performance of interpretable models. Macro average F_1-score is usually used to evaluate the model performance of unbalanced datasets, which is also widely used to evaluate the model performance in interpretable learning tasks (Fan et al., 2022; Dzikovska et al., 2012; 2013). For a set of classes D,each represented with N_d instances in the test samples, the Macro average F_1-score is given by:

$$macro \quad F_1 - score = \frac{1}{|D|} \sum_{d \in D} F_1(d) \tag{10}$$

As mentions in (Grandini et al., 2020) , since the numerators of Macro Average Precision and Macro Average Recall are formed of values in the range [0, 1], Macro-Average methods often compute an overall mean of different measures. There is no correlation between class size and the numerator weighting, which makes classes of varying sizes equivalent. This means that large classes are just as consequential as small ones. The derived measure provides a class-based evaluation of the algorithm; a high Macro-$F1$ value indicates that the algorithm performs well across all classes, while a low value indicates that some classes were poorly predicted.

Assume that all classes is equally important for users, it is possible to derive some intuitions from the equation. The macro average F_1-score is harmonistic and can reflect the degree of fit between the predicted labels by the model and the true labels in the datasets, especially for imbalanced datasets. Thus, macro average F_1-score is used to evaluate the rationality of interpretable models while the precision and recall are also reported in all experiments.

C.2 Binary sentiment analysis

[ht!]

Table 4 demonstrates the experimental results of non-pretrained models and Bert for binary sentiment analysis with variance given in the subscript.

C.3 Fine-grained sentiment analysis

Table 5 demonstrates experimental results of non-pretrained models and Bert for fine-grained sentiment analysis with variance given in the subscripts.

D The case study about model interpretability in original version

Understanding why the model generates incorrect predictions is more valuable than understanding why it generates correct predictions, especially for high-stakes domains, due to the fact that the majority of deep learning models are unable to achieve 100% accuracy in practical applications. Knowing the reasons behind the model's erroneous choices helps to minimize risks, make more informed decisions and raise the fidelity of interpretable models.

Our proposed CLK can be used to provide explanations based on various sentiment analysis datasets. In our main experiments, we conduct extensive experiments on Yelp, IMDB and Weibo datasets. Here, we we demonstrate the interpretable result from a trained model (Bert+CLK) with a real

	Models	P	R	F_1	Acc
IMDB	LR$^\diamondsuit$	52.00 $\pm_{0}$	51.98 $\pm_{0}$	51.92$\pm_{0}$	52.00$\pm_{0}$
	KNN$^\diamondsuit$	50.01$\pm_{1e-5}$	50.01 $\pm_{1e-5}$	49.87$\pm_{1e-5}$	50.01$\pm_{1e-5}$
	DT$^\diamondsuit$	52.17 $\pm_{0}$	52.10 $\pm_{0}$	51.72 $\pm_{0}$	52.11 $\pm_{0}$
	EBM$^\diamondsuit$	61.85$\pm_{0}$	61.84 $\pm_{0}$	61.83$\pm_{0}$	61.84$\pm_{0}$
	BLSTM	81.96 $\pm_{2e-5}$	81.79 $\pm_{1e-5}$	81.77 $\pm_{1e-5}$	81.79 $\pm_{1e-5}$
	BLSTM+LIME*	81.96 $\pm_{2e-5}$	81.79 $\pm_{1e-5}$	81.77 $\pm_{1e-5}$	81.79 $\pm_{1e-5}$
	BLSTM+Ours	82.93 $\pm_{1e-5}$	82.81 $\pm_{1e-5}$	82.79 $\pm_{1e-5}$	**82.81**$\pm_{1e-5}$
	BGRU	82.73 $\pm_{2e-5}$	82.57 $\pm_{3e-5}$	82.54 $\pm_{3e-5}$	82.57 $\pm_{3e-5}$
	BGRU+LIME*	82.73 $\pm_{2e-5}$	82.57 $\pm_{3e-5}$	82.54 $\pm_{3e-5}$	82.57 $\pm_{3e-5}$
	BGRU+Ours	83.30 $\pm_{2e-5}$	83.27 $\pm_{2e-5}$	83.26 $\pm_{2e-5}$	**83.27** $\pm_{2e-5}$
	Bert	83.99 $\pm_{3e-6}$	83.76 $\pm_{1e-5}$	83.73 $\pm_{1e-5}$	83.76 $\pm_{1e-5}$
	BertLIME*	83.99 $\pm_{3e-6}$	83.76 $\pm_{1e-5}$	83.73 $\pm_{1e-5}$	83.76 $\pm_{1e-5}$
	Bert+Ours	84.21 $\pm_{2e-6}$	84.13 $\pm_{6e-7}$	84.12 $\pm_{6e-7}$	**84.13** $\pm_{6e-7}$
Yelp-2	LR$^\diamondsuit$	44.19 $\pm_{0}$	50.00 $\pm_{0}$	46.92 $\pm_{0}$	88.39 $\pm_{0}$
	KNN$^\diamondsuit$	69.89 $\pm_{3e-3}$	51.81 $\pm_{2e-5}$	50.52 $\pm_{1e-4}$	87.52 $\pm_{1e-5}$
	DT$^\diamondsuit$	53.89 $\pm_{0}$	50.11 $\pm_{0}$	47.17 $\pm_{0}$	87.57 $\pm_{0}$
	EBM$^\diamondsuit$	87.81 $\pm_{0}$	51.71 $\pm_{0}$	50.18 $\pm_{0}$	88.12 $\pm_{0}$
	BLSTM	79.75 $\pm_{3e-3}$	76.97 $\pm_{1e-3}$	77.25 $\pm_{4e-3}$	90.14 $\pm_{4e-3}$
	BLSTM+LIME*	79.75 $\pm_{3e-3}$	76.97 $\pm_{1e-3}$	77.25 $\pm_{4e-3}$	90.14 $\pm_{4e-3}$
	BLSTM+Ours	83.68 $\pm_{3e-4}$	77.06$\pm_{6e-4}$	79.70$^\wr$$\pm_{2e-4}$	**91.98** $\pm_{2e-5}$
	BGRU	84.63$\pm_{8e-4}$	74.82 $\pm_{2e-4}$	78.48 $\pm_{8e-5}$	91.86 $\pm_{3e-5}$
	BGRU+LIME*	84.63$\pm_{8e-4}$	74.82 $\pm_{2e-4}$	78.48 $\pm_{8e-5}$	91.86 $\pm_{3e-5}$
	BGRU+Ours	83.44 $\pm_{3e-4}$	78.02 $\pm_{3e-4}$	80.28$^\wr$$\pm_{1e-4}$	**92.05** $\pm_{1e-5}$
	Bert	85.29 $\pm_{1e-3}$	82.88$\pm_{3e-4}$	83.89 $\pm_{1e-4}$	93.16 $\pm_{3e-5}$
	Bert+LIME*	85.29 $\pm_{1e-3}$	82.88$\pm_{3e-4}$	83.89 $\pm_{1e-4}$	93.16 $\pm_{3e-5}$
	Bert+Ours	86.53 $\pm_{2e-4}$	83.26 $\pm_{1e-4}$	84.74$^\wr$ $\pm_{2e-5}$	**93.61** $\pm_{1e-5}$

Table 4: Results of non-pretrained models and Bert for binary sentiment analysis with variance given in the subscript, where $^\diamondsuit$ denotes the glass model, LIME* denotes the explainability techniques like LIME, such as SHAP and Partial Dependence. $^\wr$ indicates the best macro F_1-score on unbalanced datasets and the bold number indicates the best Acc socre over black-box baselines.

Dataset	**Models**	**P**	**R**	$\boldsymbol{F_1}$	**Acc**
Weibo-8	LR$^\diamondsuit$	10.37$\pm_{3e-3}$	12.52$\pm_{2e-7}$	9.71$\pm_{1e-6}$	60.91$\pm_{2e-4}$
	KNN$^\diamondsuit$	22.83$\pm_{2e-3}$	13.31 $\pm_{1e-5}$	11.23 $\pm_{2e-5}$	62.93$\pm_{2e-6}$
	DT$^\diamondsuit$	22.68 $\pm_{0}$	14.21 $\pm_{0}$	12.62 $\pm_{0}$	63.23 $\pm_{0}$
	EBM$^\diamondsuit$	12.03 $\pm_{0}$	12.61 $\pm_{0}$	9.91 $\pm_{0}$	62.88 $\pm_{0}$
	BLSTM	37.39$\pm_{4e-4}$	16.08 $\pm_{1e-5}$	15.95 $\pm_{1e-4}$	42.53 $\pm_{5e-3}$
	BLSTM+LIME*	37.39$\pm_{4e-4}$	16.08 $\pm_{1e-5}$	15.95 $\pm_{1e-4}$	42.53 $\pm_{5e-3}$
	BLSTM+Ours	36.53 $\pm_{3e-3}$	16.36$\pm_{1e-4}$	16.63$^\wr$ $\pm_{1e-4}$	**45.28** $\pm_{1e-3}$
	BGRU	47.13 $\pm_{3e-3}$	16.40 $\pm_{2e-5}$	17.76$\pm_{1e-4}$	56.56 $\pm_{1e-3}$
	BGRU+LIME*	47.13 $\pm_{3e-3}$	16.40 $\pm_{2e-5}$	17.76$\pm_{1e-4}$	56.56 $\pm_{1e-3}$
	BGRU+Ours	46.72$\pm_{4e-4}$	16.96$\pm_{3e-6}$	18.43$^\wr$$\pm_{1e-5}$	**58.02**$\pm_{1e-3}$
	Bert	31.35 $\pm_{3e-5}$	29.30 $\pm_{1e-4}$	29.77$\pm_{4e-5}$	64.41 $\pm_{1e-4}$
	Bert+LIME*	31.35 $\pm_{3e-5}$	29.30 $\pm_{1e-4}$	29.77$\pm_{4e-5}$	64.41 $\pm_{1e-4}$
	Bert+Ours	36.89 $\pm_{4e-4}$	32.84$\pm_{1e-4}$	33.59$^\wr$$\pm_{1e-4}$	**65.54**$\pm_{1e-5}$
Yelp-4	LR$^\diamondsuit$	44.19$\pm_{2e-3}$	50.00$\pm_{2e-5}$	46.92$\pm_{1e-4}$	88.39$\pm_{1e-3}$
	KNN$^\diamondsuit$	69.89 $\pm_{1e-4}$	51.81$\pm_{1e-4}$	50.52$\pm_{1e-4}$	87.52$\pm_{2e-4}$
	DT$^\diamondsuit$	53.89 $\pm_{0}$	50.11 $\pm_{0}$	47.17 $\pm_{0}$	87.57 $\pm_{0}$
	EBM$^\diamondsuit$	87.81 $\pm_{0}$	51.71 $\pm_{0}$	50.18 $\pm_{0}$	88.12 $\pm_{0}$
	BLSTM	72.89$\pm_{2e-4}$	70.47$\pm_{2e-4}$	71.41$\pm_{2e-4}$	71.06$\pm_{1e-4}$
	BLSTM+LIME*	72.89$\pm_{2e-4}$	70.47$\pm_{2e-4}$	71.41$\pm_{2e-4}$	71.06$\pm_{1e-4}$
	BLSTM+Ours	73.72$\pm_{1e-4}$	70.51 $\pm_{1e-4}$	71.69$^\wr$ $\pm_{2e-5}$	**71.41**$\pm_{1e-5}$
	BGRU	71.19 $\pm_{2e-4}$	71.04$\pm_{4e-5}$	71.00$\pm_{1e-4}$	70.99 $\pm_{1e-4}$
	BGRU+LIME*	71.19 $\pm_{2e-4}$	71.04$\pm_{4e-5}$	71.00$\pm_{1e-4}$	70.99 $\pm_{1e-4}$
	BGRU+Ours	72.90 $\pm_{2e-4}$	71.17 $\pm_{4e-4}$	71.61$^\wr$ $\pm_{2e-4}$	**71.48** $\pm_{1e-4}$
	Bert	76.42 $\pm_{1e-4}$	75.81$\pm_{2e-4}$	75.89$\pm_{1e-4}$	75.43 $\pm_{1e-4}$
	Bert+LIME*	76.42 $\pm_{1e-4}$	75.81$\pm_{2e-4}$	75.89$\pm_{1e-4}$	75.43 $\pm_{1e-4}$
	Bert+Ours	77.34 $\pm_{1e-4}$	76.44 $\pm_{1e-4}$	76.77$^\wr$ $\pm_{1e-4}$	**76.18** $\pm_{1e-4}$

Table 5: Results of non-pretrained models and Bert for fine-grained sentiment analysis with variance given in the subscripts, where $^\diamondsuit$ denotes the glass model, LIME* denotes the explainability techniques like LIME, such as SHAP and Partial Dependence. $^\wr$ indicates the best macro F_1-score on unbalanced datasets and the bold number indicates the best Acc socre over black-box baselines.

example in SST-2 test data, where **SST-2**[9] is a binary sentiment analysis dataset provided by (Socher et al., 2013).

Example 2 *Consider the sequence $task_{1001}$ = "day is not a great bond movie , but it is a good bond movie , which still makes it much better than your typical bond knock-offs."*

The TPK model for this example is shown in Figure 7(a). Now we give the detail of each state in TPK. In this paper, all actions refer to get a word and classify the obtained sequence. Although they're important in sentiment analysis tasks, these actions are not the root cause of different states, so we do not label actions in the logic graph of the TPK model. Indeed, the action could be seen as the successor relation. As we mentioned in Section 3.2, for the state sets S in TPK, we construct them by the obtained sequence of DNNs such that $s_i = \{\boldsymbol{h}_{e_1}, ..., \boldsymbol{h}_{e_i}, r\} \quad (i > 0)$, where $s_i \in S$ denotes the i-th state; r denotes a propositional letter generated by the learned TPK. Since $\boldsymbol{h}_{e_i}$ is the hidden state of x_i, we utilize the corresponding tokens (words) to demonstrate the transition of states in TPK. Therefore, the information of each state in **Example 2** is as below:

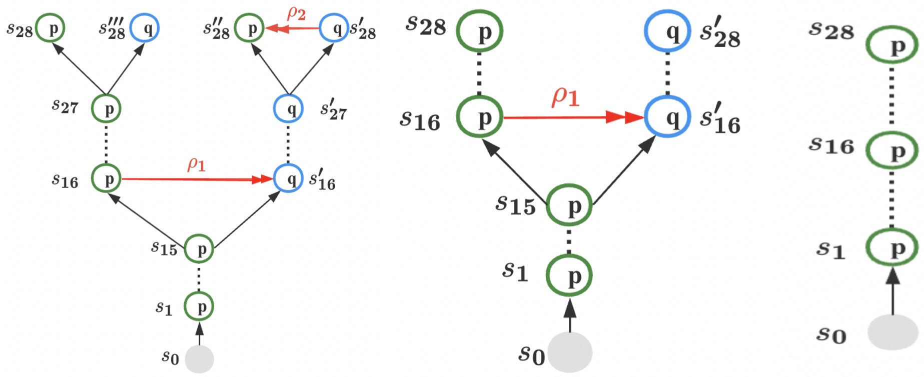
- s_0 is the start state of model
- a_1={getting a new token "day", and recognizing the obtained sequence as "negative"}
- s_1={"day", p}
- a_2={getting a new token "is", and recognizing the obtained sequence as "negative"}
- s_2={"day is", p}
- ...
- a_{15}={getting a new token "movie", and recognizing the obtained sequence as "negative"}
- s_{15}={"day is not a great bond movie, but it is a good bond movie", p}
- a_{16}={getting a new token ",", and recognizing the obtained sequence as "positive"}
- s_{16}={"day is not a great bond movie , but it is a good bond movie,", p }
- a'_{16}={getting a new token ",", and recognizing the obtained sequence as "positive"}
- s'_{16}={"day is not a great bond movie , but it is a good bond movie,", q }
- $s_{16}\rho_1 s'_{16}$
- ...
- a'_{27}={getting a new token "bond", and recognizing the obtained sequence as "positive"}
- s'_{27}={"day is not a great bond movie , but it is a good bond movie, which still makes it much better than your typical bond", q }
- a'_{28}={getting a new token "knock-offs", and recognizing the obtained sequence as "positive"}
- s'_{28}={"day is not a great bond movie , but it is a good bond movie, which still makes it much better than your typical bond knock-offs", q }
- a''_{28}={getting a new token "knock-offs", and recognizing the obtained sequence as "negative"}
- s''_{28}={"day is not a great bond movie , but it is a good bond movie, which still makes it much better than your typical bond knock-offs", p }
- $s'_{28}\rho_2 s''_{28}$

The above information of each state in TPK indicates the sentimental information of parts of a sentence and the development of a sentence's overarching sentiment from beginning to end. As a result, both the overarching sentiment and the sentiment of individual sentence components can be displayed and analyzed. And the explanations can be obtained based on these states and successor relation in TPK. Now we elaborate the explanation generating and reasoning process as follows.

Question: Why the corresponding prediction is "negative" rather than "positive"?

Explanation: Because the "knock-offs" (public announcement) appears in the following words, so the model classifies this sequence into "negative".

[9] https://nlp.stanford.edu/sentiment/



(a) (b) (c)

Figure 7: (a) shows the TPK model of **Example 2**, (b) shows the counterfactual TPK model for ρ_2, (c) shows the counterfactual TPK model for ρ_1 in **Example 2**.

As shown in Figure 7(a), we can visualize the explanation of this example in the form of TPK based on the knowledge representation in Section 3. In the diagram of TPK models, the arrow $\rightarrow$ represents successor relation R, double arrow $\twoheadrightarrow$ represents the relation ρ, and s_0 is the root of the tree structure. Let the proposition letter p denote that the sequence is recognized as "negative", proposition letter q denote that the sequence is recognized as "positive". Then $task_{1001}$ is correctly classified iff $(s'_{28} \vDash q)$, where s_i and s'_i denote the states of i^{th} word in the sequence. The interpretable results are shown in Figure 7(a), where $s_{15} \vDash Yp$. At state s_{16}, we recognise that the reason path with $\{s_0, s_1, s_3, \ldots, s_{16}\}$ contains an incompatibility because it uses the choice of "negative" and this is incompatible with it being a good movie. Thus, there is a public announcement that $s_{16}\rho_1 s'_{16}$, so we have $s_{16} \vDash \boxminus q$ and $s'_{16} \vDash \mathbb{Y}p$. Then, the reason path becomes $\{s_0, s_1, \ldots, s_{16}, s'_{16}, \ldots, s'_{27}\}$. Similarly, there is another public announcement that $s'_{28}\rho_2 s''_{28}$ at state s'_{28} with $s'_{28} \vDash \boxminus p$ and $s''_{28} \vDash \mathbb{Y}q$, ultimately the $task_{1001}$ is classified as "negative" with $s''_{28} \vDash p$. From the perspective of human knowledge, it is clear that $task_{1001}$ is incorrectly classified. Based on interpretation of TPK model, we observed that the sequence is correctly classified before s'_{27}, but there is a (wrong) public announcement at s''_{28}, resulting in the wrong prediction.

To investigate whether the explanation based on the TPK model in Figure 7(a) is trustworthy, we generate a counterfactual state for ρ_2 through a mathematical operator $do(t_0)$ interventions on s'_{28}. Then we can obtain a counterfactual TPK model as shown in Figure 7(b), where $s_{15} \vDash Yp$ and $s'_{16} \vDash \Box q$. Moreover, there is only one public announcement ρ_1 such that $s_{16}\rho_1 s'_{16}$, and the corresponding reason path is $\{s_0, s_1, \ldots, s_{16}, s'_{16}, \ldots, s'_{28}\}$. In this counterfactual case, $task_{1001}$ is classified as "positive", which indicates that the public announcement ρ_2 is accurate.

Similarly, we can also prove that ρ_1 is accurate by employing the counterfactual reasoning. Specifically, we generate counterfactual states for ρ_1 through interventions. Then we obtain another counterfactual TPK model as shown in Figure 7(c), where $s_{28} \vDash Yp$ and $s_1 \vDash \Box p$. Therefore, we can derive that $s_{16}\rho_1 s'_{16}$ and $s'_{28}\rho_2 s''_{28}$ is accurate, which further demonstrate that the explanation provided by TPK models is trustworthy.

Therefore, the model derives a wrong prediction is because there is a word "knock-offs" in $task_{1001}$, which results in $s'_{28}\rho_2 s''_{28}$; and the model can classify $task_{1001}$ correctly as "positive" without the word "knock-offs" in $task_{1001}$. Based on the above reasoning process, we can summarize that all public announcements are accurate, and the corresponding explanation is trustworthy. Finally, it should be noted that the proposed methods are not constrained to the domain of sentiment analysis, but can be used for any text classification endeavor.

E Discussion and comparison with other logic

E.1 Comparing with branching time temporal logic

TPK is different from the existing branching time temporal logic such as Computation Tree Logic (CTL) in essence (Heljanko et al., 1997). On the one hand, temporal operators in CTL are restricted to the possible future paths from a given state, and some temporal operators in CTL, such as next, globally, and until, are not available in TPK. It is worth noting that the true value of the formula in the current state in CTL will not be affected by future states, i.e., time cannot be reversed, nor can historical results be modified by returning from the future. On the other hand, public announcement is the key of TPK, i.e., a current state depends on the future and therefore may be not clear cut at the present and need future clarifications. So TPK and CTL aren't directly comparable, but there are some common ideas between these two in terms of the representation of the past and future of a state.

E.2 Comparing with epistemic logic

Talmudic logic treats states much like the quantum superposition of states and when there is a public announcement we get a collapse onto a pure state. This is similar to how the epistemic model of the agent changes after a public announcement occurs in epistemic logic (EL) and dynamic epistemic logic (DEL) (Ågotnes & Alechina, 2006). One of the main purposes of the public announcement both in TPK and epistemic logic is to update the existing model and increase its certainty. But this idea is implemented by different ways in two types of logical systems. Besides, a public announcement in EL or DEL deletes some possible worlds after an announcement, while a public announcement in TPK deletes some accessibility links after an announcement. Technically, these two approaches are similar but not equivalent.

F Proof of some equations in Knowledge acquisition

F.1 Conjugation of Dirichlet distribution and Multinomial distribution

$$Dirichlet(\boldsymbol{p}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod\limits_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} p_k^{\alpha_k - 1} \tag{11}$$

Likelihood function of multinomial distribution:

$$Multinomial(\boldsymbol{n}|\boldsymbol{p}, \boldsymbol{N}) = \binom{N}{\boldsymbol{n}} \prod_{k=1}^{K} p_k^{n_k} \tag{12}$$

Thus:

$$\begin{aligned} &\frac{Dirichlet(\boldsymbol{p}|\boldsymbol{\alpha}) Multinomial(\boldsymbol{n}|\boldsymbol{p}, \boldsymbol{N})}{\int_0^1 Dirichlet(\boldsymbol{p}|\boldsymbol{\alpha}) Multinomial(\boldsymbol{n}|\boldsymbol{p}, \boldsymbol{N}) d\boldsymbol{p}} \\ &= \frac{\prod\limits_{k=1}^{K} p_k^{\alpha_k + n_k - 1}}{\int_0^1 \prod\limits_{k=1}^{K} p_k^{\alpha_k + n_k - 1} d\boldsymbol{p}} \sim Dirichlet(\boldsymbol{p}|\boldsymbol{\alpha} + \boldsymbol{n}) \end{aligned} \tag{13}$$

We can write it as below:

$$Dirichlet(\boldsymbol{p}|\boldsymbol{\alpha}) + Multinomial(\boldsymbol{n}|\boldsymbol{p}, \boldsymbol{N}) = Dirichlet(\boldsymbol{p}|\boldsymbol{\alpha} + \boldsymbol{n}) \tag{14}$$

F.2 Proof of Equation 2

The prior distribution of θ_v is $Dirichlet(\boldsymbol{\theta_v}|\boldsymbol{\alpha})$ as below:

$$\begin{aligned} P(\boldsymbol{\theta_v}|\boldsymbol{\alpha}) &\sim (Dirichlet(\boldsymbol{\theta_v}|\boldsymbol{\alpha}) \\ &= \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod\limits_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} {\theta_v}_k^{\alpha_k - 1} \end{aligned} \tag{15}$$

Thus, the posterior distribution of θ_v is given by:

$$\begin{aligned} P(\boldsymbol{\theta_v}|\boldsymbol{n_v},\boldsymbol{\alpha}) &\sim Dirichlet(\boldsymbol{\theta_v}|\boldsymbol{(\alpha+n_v)}) \\ &= \frac{\Gamma(\sum_{k=1}^{K}(\alpha_k+n_v^k))}{\prod\limits_{k=1}^{K}\Gamma(\alpha_k+n_v^k)}\prod_{k=1}^{K}{\theta_v}^{\alpha_k+n_v^k-1} \end{aligned} \tag{16}$$

Then the topic probability distribution of X_v is given by:

$$\begin{aligned} P(\boldsymbol{z_v}|\boldsymbol{\alpha}) &= \int P(\boldsymbol{z_v}|\boldsymbol{\theta_v} P(\boldsymbol{\theta_v}|\boldsymbol{n_v},\boldsymbol{\alpha}) d\boldsymbol{\theta_v} \\ &= \int \prod_{k=1}^{K}({\theta_{v(k)}}^{n_v^k})\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod\limits_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}{\theta_{v(k)}}^{\alpha_k-1}d\boldsymbol{\theta_v} \\ &= \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod\limits_{k=1}^{K}\Gamma(\alpha_k)}\int\prod_{k=1}^{K}{\theta_{v(k)}}^{n_v^k+\alpha_k-1}d\boldsymbol{\theta_v} \\ &= \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod\limits_{k=1}^{K}\Gamma(\alpha_k)}\frac{\prod\limits_{k=1}^{K}\Gamma(\alpha_k+n_v^k)}{\Gamma(\sum_{k=1}^{K}\alpha_k)} \end{aligned} \tag{17}$$

Thus, the topic probability distribution is as below :

$$P(\boldsymbol{z}|\boldsymbol{\alpha}) = \prod_{v=1}^{V}P(\boldsymbol{z_m}|\boldsymbol{\alpha}) = \prod_{v=1}^{V}\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod\limits_{k=1}^{K}\Gamma(\alpha_k)}\frac{\prod\limits_{k=1}^{K}\Gamma(\alpha_k+n_v^k)}{\Gamma(\sum_{k=1}^{K}(\alpha_k+n_v^k))} \tag{18}$$

F.3 PROOF OF EQUATION 3

Similarly, the prior distribution of ϕ_k is a Dirichlet distribution with prior parameter β such that:

$$P(\boldsymbol{\phi_k}|\boldsymbol{\beta}) \sim (Dirichlet(\boldsymbol{\phi_k}|\boldsymbol{\beta}) = \frac{\Gamma(\sum_{m=1}^{M}\beta_m)}{\prod\limits_{m=1}^{M}\Gamma(\beta_m)}\prod_{m=1}^{M}{\phi_k}_m^{\beta_m-1} \tag{19}$$

And, the posterior distribution of ϕ_k is given by:

$$\begin{aligned} P(\boldsymbol{\phi_k}|\boldsymbol{n_k},\boldsymbol{\beta}) &\sim Dirichlet(\boldsymbol{\phi_k}|\boldsymbol{(\beta+n_k)}) \\ &= \frac{\Gamma(\sum_{m=1}^{M}(\beta_m+n_k^m))}{\prod\limits_{m=1}^{M}\Gamma(\beta_m+n_k^m)}\prod_{m=1}^{M}{\phi_k}_m^{\beta_m+n_k^m-1} \end{aligned} \tag{20}$$

So we can obtain the word probability distribution of the k-th topic by:

$$\begin{aligned}
P(\boldsymbol{x_k}|\boldsymbol{\beta}) &= \int P(\boldsymbol{x_k}|\boldsymbol{\phi_k})P(\boldsymbol{\phi_k}|\boldsymbol{\beta})d\boldsymbol{\phi_k} \\
&= \int \prod_{m=1}^{M} (\phi_{k(m)}{}^{n_k^m} \frac{\Gamma(\sum_{m=1}^{M} \beta_m)}{\prod\limits_{m=1}^{M} \Gamma(\beta_m)} \prod_{m=1}^{M} \phi_{k(m)}{}^{\beta_m - 1} d\boldsymbol{\phi_k} \\
&= \frac{\Gamma(\sum_{m=1}^{M} \beta_m)}{\prod\limits_{m=1}^{M} \Gamma(\beta_m)} \int \prod_{m=1}^{M} \phi_{k(m)}{}^{n_k{}^m + \beta_m - 1} d\boldsymbol{\phi_k} \\
&= \frac{\Gamma(\sum_{m=1}^{M} \beta_m)}{\prod\limits_{m=1}^{M} \Gamma(\beta_m)} \frac{\prod\limits_{m=1}^{M} \Gamma(\beta_m + n_k^m)}{\Gamma(\sum_{m=1}^{M} \beta_m + n_k^m)} \qquad (21)
\end{aligned}$$

Thus, when given the topic the word distribution is calculated by:

$$\begin{aligned}
P(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\beta}) &= \prod_{k=1}^{K} P(\boldsymbol{x_k}|\boldsymbol{\beta}) \\
&= \prod_{k=1}^{K} \frac{\Gamma(\sum_{m=1}^{M} \beta_m)}{\prod\limits_{m=1}^{M} \Gamma(\beta_m)} \frac{\prod\limits_{m=1}^{M} \Gamma(\beta_m + n_k^m)}{\Gamma(\sum_{m=1}^{M} (\beta_m + n_k^m))} \qquad (22)
\end{aligned}$$

F.4 PROOF OF EQUATION 4

Finally, the joint probability distribution for topic-word is as follows:

$$\begin{aligned}
P(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}) &= P(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\beta})P(\boldsymbol{z}|\boldsymbol{\alpha}) \\
&= \prod_{k=1}^{K} \frac{\Gamma(\sum_{m=1}^{M} \beta_m)}{\prod\limits_{m=1}^{M} \Gamma(\beta_m)} \frac{\prod\limits_{m=1}^{M} \Gamma(\beta_m + n_k^m)}{\Gamma(\sum_{m=1}^{M} (\beta_m + n_k^m))} \\
& \prod_{v=1}^{V} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod\limits_{k=1}^{K} \Gamma(\alpha_k)} \frac{\prod\limits_{k=1}^{K} \Gamma(\alpha_k + n_v^k)}{\Gamma(\sum_{k=1}^{K} (\alpha_k + n_v^k))} \qquad (23)
\end{aligned}$$