Multiview Commonsense Reasoning using LLMs for Understanding Crime Drama Series

Muhammad Abdullah Zia[1], Sameen Mansha *[2], and Faisal Kamiran[1]

[1] Department of Computer Science, Information Technology University of the Punjab, 54600 Lahore, Pakistan
abdullahzia510@gmail.com; faisal.kamiran@itu.edu.pk
[2] KTH Royal Institute of Technology, 164 40 Kista, Stockholm, Sweden
sameen@kth.se

Abstract. Crime Scene Investigation (CSI) is a forensic crime-based series where perpetrators often try to hide their motives to cover up murders. In contrast, investigators trace pieces of evidence to spot culprits. Recognizing the original character played by a particular speaker (i.e., perpetrator, investigator, and suspects), corresponding to any CSI-based dialogue, using textual conversations is challenging. Existing approaches do not use deep multiview learning for processing multiview commonsense-based Knowledge Graph (KG). Our proposed approach, RiMCR, first applies Siamese BERT-Networks (SBERT) to learn sentence structure. We process sixteen multiview relations of commonsense-based knowledge graph $ATOMIC^{20}_{20}$ through COMET(BART). A dual-view deep network architecture based on independent stacked LSTMs with a self-attention mechanism infuses sequential patterns into sentence and common-sense-based features. Lastly, we concatenate four types of encoded features before passing through the decoder to solve binary and multiclass classification problems. An extensive comparison with sequence models and Large Language Models (LLMs) validates the judiciousness of RiMCR.

Keywords: Commonsense based Knowledge Graph, Deep Multiview Learning, Large Language Models, Dual View Network, Crime Drama Understanding.

1 Introduction and Motivation

CSI [10] is a forensic crime-based series that premiered from October 2000 to September 2015. The length of an episode, on average, is 45 minutes. The perpetrator murders a victim at the beginning of every episode. A team of investigators hired by the Las Vegas Police Department interrogates all suspects and related people to identify the murderer in every case. Sometimes, interrogators resolve multiple independent murder cases in a single episode.

A crime scene investigator possesses a keen understanding of the criminal justice system, scientific observations, and methods, primarily related to the

* Corresponding author

forensic sciences for determining the type of weapon, placement of victims and suspects, time of crimes, etc. Investigators excerpt every possible information from crime scenes, forensic reports, and interrogatory dialogues. The mystery is resolved at the end of the episode; the culprit is found guilty, and innocent suspects are acquitted. Hence, understanding criminals' motivation or mental condition is tricky, and it often requires skills in criminal psychology.

CSI dialogues can be classified into two categories: (1) Interrogation and (2) Scene description. The interrogatory dialogues, spoken by the investigators, suspects, and perpetrators, have intricate semantic relationships. Similarly, scene descriptions are valuable, entailing how any case is picturized (e.g., atmosphere, positioning of victims, crime scenes, etc). Drama comprehension is an interesting application of natural language understanding.

2 Challenges and Contributions

Multiview data has become one of the primary data types where divergent feature sets describe an image, object, or event. A feature set representing similar variables, such as an angle or crop of an image, is called a view. An image containing miscellaneous self-sufficient feature types (e.g., texture, color, or shape) must be processed independently through multiple views. These views can also contain inconsistent and redundant information. Deep multiview learning is successfully adopted to study diverse and complementary views through multiple self-contained units in an end-to-end manner. In the last decade, it has been adopted for numerous healthcare applications, such as analyzing different angles of an X-ray image for cancer detection [11]. Besides continuous data types (e.g., image, audio, and video processing), in recent years, deep multiview learning has been applied to disentangle twisted technicalities from a wide range of discrete data types, such as multilingual NLP or graph types [29]. For multiview graphs, unsupervised and self-supervised clustering-based techniques are also studied [22].

In this paper, we identify the role of the speaker, i.e., if a dialogue has been spoken by the perpetrator, investigator, or others; also, if it is a scene description. Even human guesses for role identification, especially distinguishing between perpetrator and suspects, during the CSI screening are often inaccurate. Hence, CSI is suspenseful and situation-dependent, where actors do not express similar emotions through specific words, perpetrators frequently lie, suspects can misguide, and forensic reports are to be interpreted correctly. CSI dialogues are intrinsically multiview, each view having different semantics. Thus, designing a deep multiview, common-sense reasoning-based approach for dialogue-specific role identification is non-trivial.

In this paper, we will deal with the following challenges related to the adaptation of deep multiview learning for CSI data:

(C1) Representation and Feature Extraction: a) How to prepare sentence-structure-based features? b) How do we derive commonsense reasoning-based features to understand social interactions, physical entities, or events? **(C2)**

Table 1: 16 commonsense relations of $ATOMIC^{20}_{20}$ are used in this paper.

Social-Interactions				**Physical-Entity**		**Event-Centered**	
Relation	*Size*	*Relation*	*Size*	*Relation*	*Size*	*Relation*	*Size*
oWant	94,548	xNeed	128,955	**ObjectUse**	165,590	**IsAfter**	22,453
oEffect	80,166	xAttr	148,194	**AtLocation**	20,221	**HasSubEvent**	12,845
oReact	67,236	xEffect	115,124	**MadeUpOf**	3,345	**isBefore**	23,208
		xReact	81,397	**HasProperty**	5,617	HinderedBy	106,658
		xWant	135,360	CapableOf	7,968	Causes	376
		xIntent	72,677	Desires	2,737	xReason	334
				Not Desires	2,838	isFilledBy	33,266

Alignment: To align derived feature representations, we must consider their complementarity, long-range dependencies, and ambiguities. To do so, a) suitable independent ML algorithms in different views are deployed. b) Depending on the nature of features derived from multiple views, their respective sizes should be stabilized. **(C3)** Fusion: How can multiview features from different views and original embeddings of LLMs be integrated? **(C4)** Optimization: Which optimization strategies in deep learning should be practiced? By tackling the discussed challenges, our main contribution is introducing *Role Identification using Multiview Commonsense Reasoning*(RiMCR):
(1) RiMCR learns sentence-level features through SBERT and stores in $\mathcal{SBFD}$. It processes 16 multiview relations of commonsense-based graph $ATOMIC^{20}_{20}$ through the COMET(BART) encoder and store them in $\mathcal{CSFD}$.
(2) For every dialogue, it accesses features from $\mathcal{SBFD}$ or $\mathcal{CSFD}$ and passes them through a dual-view deep network. Each view contains independent stacked LSTMs with the self-attention mechanism. It concatenates LLM original embeddings from $\mathcal{SBFD}$ and $\mathcal{CSFD}$ with transformer-based sequentially infused features derived through two separate views. A decoder with different loss functions is applied to solve binary or multiclass classification problems.
(3) We label a publicly available dataset to meet the requirements for dialogue-specific role identification [5,14]. A robust comparison of RiMCR with state-of-the-art sequence models and finetuned LLMs, to solve binary and multiclass problems, proves the significance of our work.

The rest of the paper is organized as follows. Section 3 discusses the related work. Section 4 defines the problem setting and notations. Section 5 describes methodology. Section 6 presents data statistics, and experimental results. Section 7 reviews literature on CSI dataset. We conclude with future work in Section 8.

3 Related work

3.1 Commonsense Based KGs

KGs are information networks that connect two related head and tail entities via relationship edges to create a triplet. Commonsense-based knowledge bases contain triplets, facts, and descriptions about daily life to perform tasks that require wisdom. The commonsense-based datasets are crafted through crowd-sourcing, language models, or manual approaches and adopted for inference,

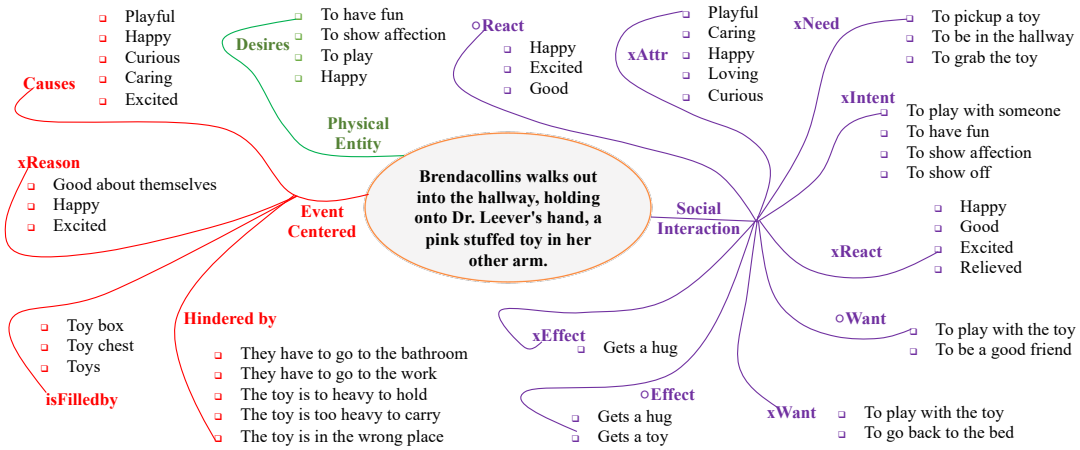


Fig. 1: Given a scene description, $ATOMIC^{20}_{20}$ generates the following graph.

question answering, knowledge base completion, etc. ConceptNet was introduced as a commonsense-based KG that connected words and phrases via labeled edges to create assertions. ConceptNet has static concepts, representing crowdsourced data from the Open Mind commonsense project [19]. To improve ConceptNet representation based on static concepts, *ATOMIC* (An Atlas of Machine Commonsense for If-Then Reasoning) introduced triplets having sequential knowledge, such as unobserved causes and events, to elaborate pre and post-situations [26]. It was organized as if-then relation types in the form of textual descriptions of inferential conditions as: (1) If event then mental state, (2) If event then event, (3) If event then persona. *ATOMIC* has more than 877,000 instances of inferential knowledge and nine if-then relation types. It mainly focused on creating relations between causes and effects, agents and themes, voluntary and involuntary events, and actions and mental states.

Later, its advanced version $ATOMIC^{20}_{20}$, having 1.33 million tuples with 23 relations was introduced [17]. Table 1 shows its 23 relations can be divided into three categories: (1) social interactions; (2) physical entities; and (3) event-centered. The size of each relation specifies relevant number of triplets, such as in the case of (**xNeed**) relation, 128,955 inferential knowledge based assertions are available. We use 16 relations from Table 1 mentioned in black for our experiments and discard relations highlighted in blue.

3.2 Multiview KG Relations in $ATOMIC^{20}_{20}$ Related to CSI

Given any phrase, $ATOMIC^{20}_{20}$ generates a set of relations and relevant tales [17]. We input a CSI scene description, "Brendacollins walks into the hallway,..", to $ATOMIC^{20}_{20}$ and show the output graph in Figure 1. Belonging to the category of social interactions, **xIntent** reveals that the person Brendacollins wants to play with someone and have fun. Holding onto Dr. Leever's hand justifies

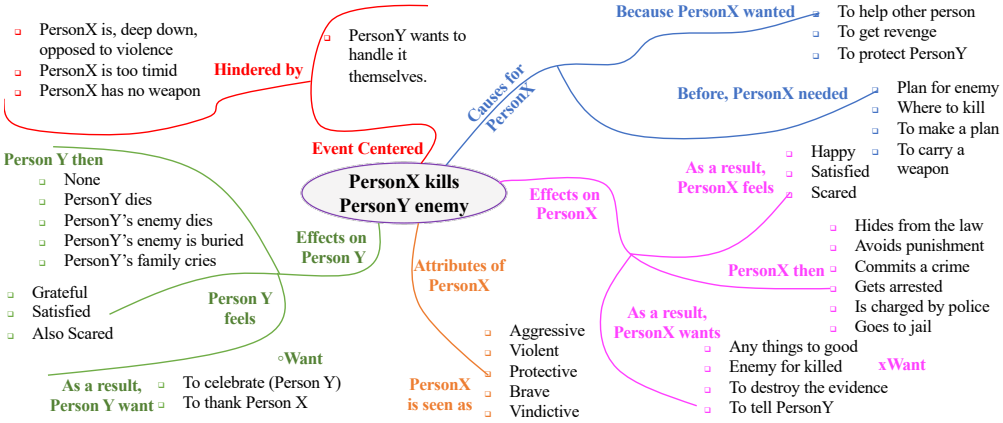


Fig. 2: Given an interrogatory dialogue, $ATOMIC^{20}_{20}$ generates graph.

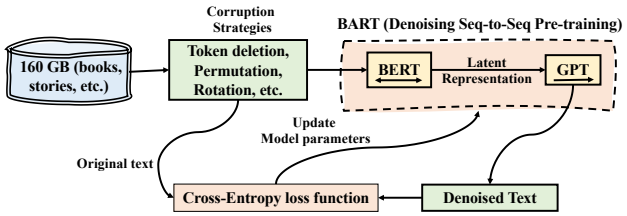


Fig. 3: Pretraining of BART-Large for reconstructing text [18].

showing affection and possible intentions of showing off to others. The relation (**xAttr**) is connected to playful, caring, happy, loving, and curious tail entities. The following events can be (**Hindered**) if they have to go to the bathroom or work; the toy is heavy to hold, carry, or placed in the wrong spot.

Figure 2 successfully comprehends all the possible perspectives of the prompt *Person X kills Person Y enemy*, where (**xWant**) and (**oWant**) are postcondition reactions of "PersonX" and "PersonY", respectively. (**xWant**) and (**oWant**) are social actions that may occur after the event: PersonX may "destroy the evidence" and Y may "celebrate" and "thank PersonX" in response. The generated graph nodes are similar to possibilities the human audience may understand during CSI show screening. Both derived graphs are fruitful for inferring emotions, reactions, and their influence on past, present, and future actions.

3.3 COMET(BART) is BART-Large Finetuned Using $ATOMIC^{20}_{20}$

BART-large [18] is composed of two main components: a bidirectional encoder (BERT) and a unidirectional left-to-right auto-regressive decoder (GPT). It processes a corrupt document to recover the original one through denoising autoencoders. The encoder learns the word embeddings by corrupting the original input

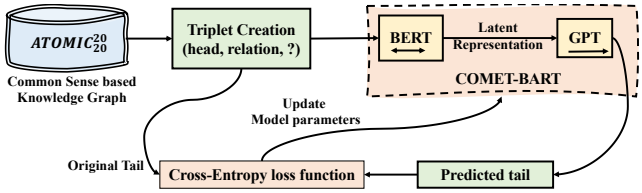


Fig. 4: COMET(BART) is BART-Large Finetuned for tail generation [17].

via random noisy transformations, e.g., token masking, deletion, infilling, permutation, and rotation. The autoregressive decoder regenerates text using a beam search algorithm and minimizes a cross-entropy-based loss function to regenerate the original document. The online available version of BART-large [1] has been pre-trained on 160 GB of textual data scrapped from textbooks, news articles, Wikipedia, and stories. It contains 406 million parameters, 24 layers, 1024 dimensional hidden states, and 16 self-attention-based heads. The pre-training steps in [18] are elaborated in Figure 3.

In COMET(BART) [17], $ATOMIC^{20}_{20}$ is used to funetune BART-large (see Figure 4). Through passing head and respective relation in Table 1 as input, BART-large is finetuned for the task of the tail generation. Its encoder (BERT) processes an incomplete tuple containing a subject and relation, while the decoder (GPT) generates tail phrases. Then, COMET(BART) parameters are updated to reduce the distance between the original and predicted tail:

4 Problem Formulation

Let $\mathcal{X} = \{x_i \| i = 1, 2, 3, \ldots\ldots s\}$ is a dataset containing s dialogues, whereas each sentence x_i is either spoken by the perpetrator, investigator, or other people; or it is a scene description. "Other" could be suspects, victims, or any other person, however, they are neither killers nor investigators. Any x_i always belongs to a single class, e.g., perpetrator and investigator are never the same, also the scene descriptions are not uttered by the drama actors. We aim to solve the following binary (Problem. 1) and multi-class (Problem. 2) classification tasks. The labels of dataset $\mathcal{X}$ are defined for s dialogues accordingly:

Problem 1. We prepare a labeled dataset $\mathcal{X}_B = \{x_i, b(x_i)\}_{i=1}^{s}$ whereas $s = \|\mathcal{X}\|$, we consider a binary classification problem having labels $\mathcal{B} = \{B^+, B^-\}$ such as $b(x_i) \leftarrow \{1, 0\}$. The dialogues in $\mathcal{X}_B$, are distinguished between those spoken by: (i) B^+, i.e., perpetrator, or (ii) B^-, i.e., belonging to the any-other source (i.e., investigator, others, and scenes). We assume that $b(x_i) := 1$ implies that the sentence is spoken by the perpetrator, whereas a zero value suggests otherwise.

Problem 2. We prepare a labeled dataset $\mathcal{X}_M = \{x_i, m(x_i)\}_{i=1}^{s}$, where x_i belongs to any of the following classes: $\mathcal{M} = \{Perpetrator, Investigator, Others, Scenes\}$. The target ground truth $m(x_i)$ defines the respective class labels for x_i in $\mathcal{M}$.

```
from sentence_transformers import SentenceTransformer
sbert_model = SentenceTransformer("roberta-large")
print(sbert_model.encode("Person X kills Person Y").size)          #Returns 1024
print(sbert_model.encode("Brendacollins walks into the hallway,
holding onto Dr. Leever's hand, ").size)                           #Returns 1024
```

Fig. 5: For two different sentences, SBERT generates embeddings of same size.

5 Methodology

In Section 5.1, we discuss how sentence and commonsense-based features for all dialogues in $\mathcal{X}$ are acquired and stored in $\mathcal{SBFD} \in \mathbb{R}^{s \times 1024}$ and $\mathcal{CSFD} \in \mathbb{R}^{s \times n}$. Given ith index of specific x_i, Section 5.2 attains embeddings from $\mathcal{SBFD}$ and $\mathcal{CSFD}$ to pass through a dual view network containing stacked LSTMs with an attention mechanism. Section 5.3 describes a decoder that predicts the final label for $\mathcal{X}_B$ and $\mathcal{X}_M$ to solve Problem.1 and Problem.2.

5.1 Representation and Feature Extraction

Sentence Level Feature Extraction SBERT [25] applies a pooling operation, through a siamese structured network, to learn sentence-level representations, from the embeddings derived by BERT or RoBERTa. In this way, SBERT reduces the embedding sizes and computational complexity of BERT-based architectures. Figure 5 represents the SBERT code snippet for two different sentences as input. For both sentences, SBERT based on RoBERTa-Large, generates embeddings of size 1024. We input x_i from $\mathcal{X}$ to sentence transformer model that first uses the RoBERTa-Large to calculate its embeddings. Then, it applies pooling on estimated embeddings to gain sentence-level embeddings of size 1024:

$$\mathcal{SBFD}[i] \leftarrow SBERT(x_i) \tag{1}$$

In this way, we create a database $\mathcal{SBFD} \in \mathbb{R}^{s \times 1024}$ containing embeddings of size 1024 for all sentences in $\mathcal{X}$. We will input ith index of any x_i in $\mathcal{X}$ to load $\mathcal{SBFD}$ respective sentence level embeddings in Sections 5.2 and 5.3.

Multi View Commonsense Feature Extraction We deploy COMET(BART), demystified in Figure 4, to acquire embeddings of every dialogue x_i in $\mathcal{X}$. Here, we deploy a multiview learning-based pipeline for x_i and elaborate with Figure 6 in two stages.

For a *single view*, we extract commonsense based features for specific x_i w.r.t one relation in $\mathcal{U} = \{u_r \| r = 1, 2, 3, \ldots\ldots, nr\}$. A supplementary sentence $\tilde{x}$ is formed, by concatenating x_i and selected relation (u_r). Then, $\tilde{x}$ is forwarded as input to COMET(BART) that generates tokens. It returns embeddings $\mathcal{CS} \in \mathbb{R}^{id \times \tilde{n}}$ where large unique tokens lead to generate a huge number of embeddings.

For procuring sentence-level commonsense features, we iterate over all embeddings w.r.t token ids to summarize them in one $\tilde{n}$ dimensional vector through the

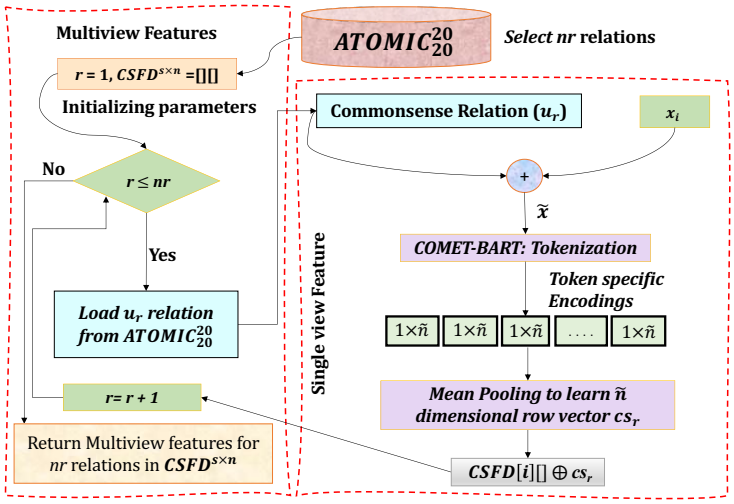


Fig. 6: Extracting multiview commonsense feature for nr relations.

mean pooling operation. We loop over token ids indices (i.e., $j = 1, 2, \ldots, \ldots, id$; $e = 1, 2, \ldots, \ldots, \tilde{n}$) for each $CS[e]$, to store embeddings in $cs_r \in \mathbb{R}^{\tilde{n}}$:

$$cs_r[e] = \frac{1}{id} \sum_{j=1}^{id} \mathcal{CS}[j][e] \tag{2}$$

For *multi view* learning, we derive commonsense-based features for x_i w.r.t. nr relations in $\mathcal{U}$, and store on ith index of $\mathcal{CSFD} \in \mathbb{R}^{s \times n}$. We repeat *single view* learning process for iterating over all relations, to estimate relation-specific features of x_i. We concatenate $\tilde{n}$ dimensional relation-specific vectors horizontally, to store in one vector $\mathcal{CSFD}[i]$ of size $n = nr \times \tilde{n}$. In this way, for s sentences in $\mathcal{X}$ we store features sequentially in $\mathcal{CSFD} \in \mathbb{R}^{s \times n}$. Given ith index of x_i from $\mathcal{X}$, we will retrieve commonsense features from $\mathcal{CSFD}$ in Sections 5.2 and 5.3.

5.2 A Dual View Network for Infusing Quintessential Sequential Patterns

In the following, Figure 7 describe how sentence-level and common-sense-specific features are passed independently through a dual view deep network containing independent stacked-LSTMs with attention mechanism. LSTM [15] is an advanced form of artificial recurrent neural network for capturing long-range sequential dependencies. In stacked-LSTMs, there can be multiple layers of LSTMs in a stacked form, and each layer can contain various LSTM cells. The output of the first layer acts as input to the next layer.

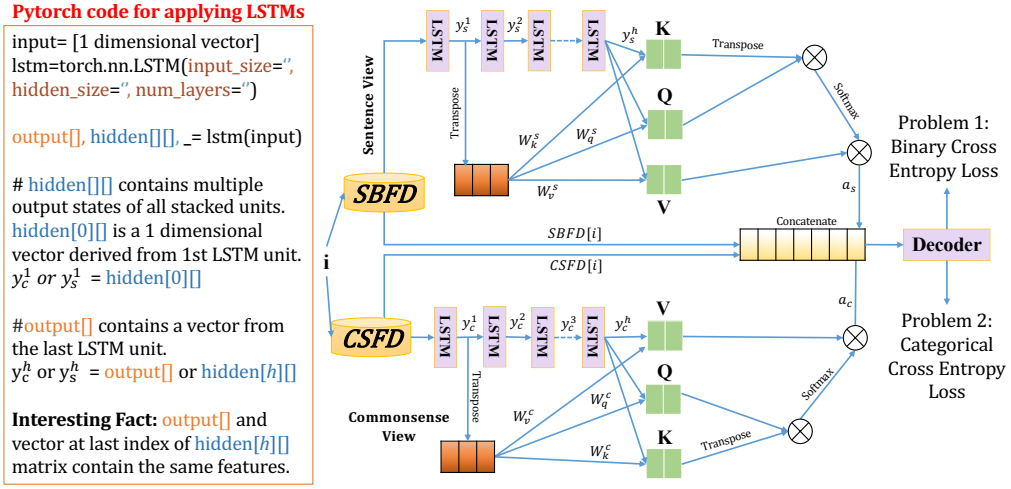


Fig. 7: Overall Framework of RiMCR with Pytorch example code snippet.

Sentence View Given ith index of x_i, we access $\mathcal{SBFD}[i]$ of 1024 dimension, and employ stacked LSTMs having h layers to process it. The LSTM units input a one dimensional vector of size 1024. The output states: $y_s \in \{y_s^1, y_s^2, y_s^3, ..., y_s^{h-1}, y_s^h\} \in \mathbb{R}^{1\times d}$ for each LSTM unit are of d dimension. Every layer of LSTM returns a vector of d dimension where we only use output from the first and last layer:

$$\begin{aligned} y_s^1 &= SBFD\_LSTM_1(\mathcal{SBFD}[i]), \\ &\text{.....}, \\ y_s^h &= SBFD\_LSTM_h(y_s^{h-1}), \end{aligned} \tag{3}$$

Self Attention processes diverse features to concentrate on important information and ignore irrelevant components [28]. We apply self attention mechanism on the outputs of first and last units of stacked-LSTMs (y_s^1 and y_s^h). It is calculated using relevant query, key, and value matrices: $\mathcal{Q}_s = W_q^s \times y_s^h; \mathcal{K}_s = W_k^s \times y_s^h; \mathcal{V}_s = W_v^s \times y_s^h$ to apply:

$$softmax\left(\frac{\mathcal{Q}_s \mathcal{K}_s^T}{\sqrt{q_s}}\right).\mathcal{V}_s \tag{4}$$

The scaling factor $\sqrt{q_s}$ is used to reduce the large magnitude of dot product, whereas $W_q^s \in \mathbb{R}^{d\times 1}; W_k^s \in \mathbb{R}^{d\times 1}; W_v^s \in \mathbb{R}^{d\times 1}$:

$$\mathcal{A}_s \in \mathbb{R}^{d\times d} = softmax\left(\frac{(W_q^s \times y_s^h).(W_k^s \times y_s^h)^T}{\sqrt{q_s}}\right).(W_v^s \times y_s^h) \tag{5}$$

Three weight matrices W_q^s, W_k^s, W_v^s are initialized with the transpose of y_s^1 to create a $d \times 1$ vector. The output matrix $\mathcal{A}_s \in \mathbb{R}^{d \times d}$ is passed through the mean pooling operation, to condense the latent representation. We loop over d dimensions ($i.e., j = 1, 2, \ldots, \ldots, d; e = 1, 2, \ldots, \ldots, d$) for each $a_s[e]$, to summarize an input row vector $a_s \in \mathbb{R}^d$:

$$a_s[e] = \frac{1}{d} \sum_{j=1}^{d} \mathcal{A}_s[j][e] \tag{6}$$

Commonsense View We access ith index of $\mathcal{CSFD}[i]$ to attain an n dimensional feature vector. We forward it to h layered stacked LSTMs where output states are depicted as $y_c \in \{y_c^1, y_c^2, y_c^3, ..., y_s^{h-1}, y_c^h\} \in \mathbb{R}^{1 \times p}$.

$$\begin{aligned} y_c^1 &= CSFD\_LSTM_1(\mathcal{CSFD}[i]), \\ &\ldots\ldots\ldots, \\ y_c^h &= CSFD\_LSTM_h(y_c^{h-1}), \end{aligned} \tag{7}$$

We apply self attention on the outputs of LSTM's first and last units (i.e., y_c^1 and y_c^h). The relevant matrices are $\mathcal{Q}_c = W_q^c \times y_c^h, \mathcal{K}_c = W_k^c \times y_c^h, \mathcal{V}_c = W_v^c \times y_c^h$:

$$\mathcal{A}_c \in \mathbb{R}^{p \times p} = softmax\left(\frac{(W_q^c \times y_c^h).(W_k^c \times y_c^h)^T}{\sqrt{q_c}}\right).\left(W_v^c \times y_c^h\right) \tag{8}$$

Respective weight matrices $W_q^c \in \mathbb{R}^{p \times 1}; W_k^c \in \mathbb{R}^{p \times 1}; W_v^c \in \mathbb{R}^{p \times 1}$ are initialized with the transpose of y_c^1, and $\sqrt{q_c}$ is scaling factor. We perform mean pooling on $\mathcal{A}_c$ to convert it to a row vector. We loop over p dimensions ($i.e., j = 1, 2, \ldots, \ldots, p; e = 1, 2, \ldots, \ldots, p$) for each $a_c[e]$, to derive $a_c \in \mathbb{R}^p$:

$$a_c[e] = \frac{1}{p} \sum_{j=1}^{p} \mathcal{A}_c[j][e] \tag{9}$$

5.3 Decoder

After encoding complex dependencies between dialogues, commonsense reasoning, and respective time series patterns through a dual view network, we model nonlinear interactions from the encoded information. We gauge the role of the speaker of x_i by concatenating four type of features: $\mathcal{SBFD}[i]$ (see Equation 1), a_s (see Equation 6), $\mathcal{CSFD}[i]$ (see Equation 2), and a_c (see Equation 9):

$$z_0 \in \mathbb{R}^f = \mathcal{SBFD}[i] \oplus a_s \oplus \mathcal{CSFD}[i] \oplus a_c \tag{10}$$

Then, z_0 is passed to a deep neural network based decoder with g layers, where $f = 1024 + d + n + p$:

$$\begin{aligned} z_1 &= GELU(W_{g1}z_0 + e_{g1}) \\ &\text{.....,} \\ z_{g-1} &= GELU(W_{g-1}z_{g-2} + e_{g-1}) \end{aligned} \tag{11}$$

where $z_1, z_2, \ldots z_{g-1} \in \mathbb{R}^f$ are the deep layers; $W_{g1}, W_{g2}, \ldots W_g \in \mathbb{R}^{f \times f}$, and $\mathbf{e}_{g1}, \mathbf{e}_{g2}, \ldots \mathbf{e}_g \in \mathbb{R}^f$ are the weights and biases; in their corresponding intermediate hidden layers. In the deep layer, we choose GELU as an activation function, while on the last layer, another relevant activation function is chosen. We use the sigmoid activation function in the case of a binary-class experiment, and for multi-class experiments, we use the softmax activation function. We minimize the following binary cross-entropy based loss function, for a batch of CSI dialogues in $\mathcal{X}_B$ where $b(x_i)$ is the actual label, and $\hat{b}(x_i)$ is predicted by RiMCR (Problem.1):

$$\begin{aligned} \hat{b}(x_i) &= sigmoid(W_g z_{g-1} + e_g) \\ \mathcal{L}_B &= -\sum_{x_i \in \mathcal{X}_B} b(x_i) log(\hat{b}(x_i)) + (1 - b(x_i)) log(1 - \hat{b}(x_i)) \end{aligned} \tag{12}$$

We minimize $\mathcal{L}_M$ based on $\mathcal{M}$ number of classes, where $m(x_i) \in \mathcal{X}_M$ is the actual label, and $\hat{m}(x_i)$ is predicted by RiMCR (Problem.2):

$$\begin{aligned} \hat{m}(x_i) &= softmax(W_g z_{g-1} + e_g) \\ \mathcal{L}_M &= -\sum_{x_i \in \mathcal{X}_M} m(x_i) log(\hat{m}(x_i)) + (1 - m(x_i)) log(1 - \hat{m}(x_i)) \end{aligned} \tag{13}$$

Note that we only access SBERT and COMET(BART) embeddings through Equation 1 and 2 in a zero-shot manner, and their gradients are not updated here. We update the gradients of the dual view network in Figure 7 containing stacked LSTMs, attention, and decoder. We initialize and train different RiMCR models that do not share any trainable weight metrices to solve both problems.

6 Experimentation and Results

The following section discusses CSI dataset characteristics, implementation details, and comparisons with state-of-the-art techniques.

Season	**Episodes**	**Cases**	*Killer* (B^+)	*Investigator*	*Others*	*Scene*	B^-	**Total**
1	5	8	287	2494	741	1927	5162	5449
2	6	8	289	2951	855	2184	5990	6279
3	7	14	505	3196	1048	3061	7305	7810
4	10	13	539	3900	1689	3094	8683	9222
5	10	14	693	3796	2214	3661	9671	10364
Total	39	57	2313	16337	6547	13927	36811	39124

Table 2: Labeled Dataset Statistics to solve Problem.1 and Problem.2.

6.1 Available Dataset

We download a textual dataset prepared by human annotators based on 39 episodes of CSI from here [5,14]. It is created to understand drama and answer the question "If the killer is mentioned in a sentence?". In this dataset, 39 episodes are randomly selected from the first 5 seasons of 59 criminal cases. Each dialogue is divided into words, and each word is labeled if that word mentions the killer through attribute "killer_gold". For each dialogue, sentence ID, case ID, speaker name, start and end time of spoken word, and human random guesses (If the speaker is a killer?) are available. The speaker's name is set to "none" for the scene descriptions in the drama.

6.2 Dataset Labeling and Preparation

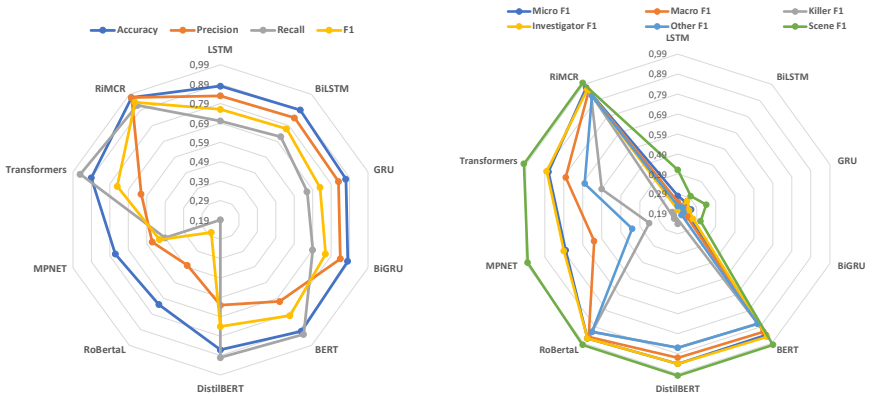
We can not identify role of speaker using available attributes in the downloaded dataset. The speaker's name is valuable in identifying scenes and unique names. We consider sentences with null speaker IDs as scenes. We manually annotate each sentence to categorize dialogues as spoken by the perpetrator or any others. We validate the correctness of labeling by watching CSI or reading transcripts from Foreverdreaming [6] and Wikipedia [10]. We dropped two cases that were not a good fit for our experiments. In dropped cases, (i) the killer was a kid who accidentally killed his brother, and (ii) the victim committed suicide. Table 2 shows there is a total of 39,124 dialogues, where killer speaks 2313 dialogues. We select 70% dialogues from each case for training, 10% for validation, and 20% as testing data through stratified sampling.

6.3 Implementation and Hyper Parameters

We perform our experiments using Quadro-P6000 GPU (24 GB) with 480 GB RAM, and 32 processors. We use Pytorch, Hugging face [7], and COMET(BART) [3]. Piling up layers in such deep architectures changes the input distribution, slows convergence, and increases training error. Batch normalization with dropout is applied to mitigate the discussed ill effects and prevent overfitting. We use Adam optimizer with mini-batches of size [8,16,32,64,128]. Weight initialization for stacked LSTMs is done using Xavier initialization. The decoder's weights are initialized using Kaiming-He. Hyperparameter tuning using grid search is done with learning rates $[1e^{-5}, 1e^{-6}]$; depth of the layers in stacked LSTMs [1-6]; size of hidden units and output vector of stacked LSTMs [32,64,128,256,512,1024]; depth of hidden layers in decoder [1-3], and dropout ratio [0.15,0.2,0.25,0.5,0.6,0.8]. For procuring commonsense features, $\tilde{n}$ is set in ranges of [512,896,1024]. We use 250 and 500 epochs with early stopping having patience of five epochs.

6.4 Performance Comparison

We compare RiMCR with the following independently trained methods:
Recurrent Network: We use LSTM [15], BiLSTM [27], GRU [12], BiGRU [27].



(a) Problem. 1 (b) Problem. 2

Fig. 8: Comparison of RiMCR with state-of-art models.

Transformers: We finetune transformer-based language models:
BERT [2] A bidirectional LLM pre-trained using masked language modeling and next sentence prediction. **DistilBERT [4]** Small-sized BERT with fewer parameters to complete the same tasks faster. **RoBertaL [9]** Improved BERT-large through byte-pair encoding, sentence-level pretraining, and removal of the next-sentence pretraining objective. **MPNET [8]** Leveraged permutated language modeling, through two stream self-attention and position compensation. **Transformers [28]** Positional encodings through self-attention mechanism.

Figure 8 portrays the results of all models for the Problem. 1 and Problem. 2. Eight recurrent, ten transformer and two RiMCR-based models are initialized to solve both problems. MPNET and RoBertaL report significantly lower scores than others when finetuned for Problem. 1. Sequence models reached the declining scores for multiclass classification, whereas RoBertaL achieved enhanced scores. BERT and DistilBERT produce stable patterns for both problems. RiMCR's highest scores for both problems yield it as the winner.

7 Literature Review on CSI Dataset [14]

A deep multi-view architecture trained recurrent neural networks of correlational GRU cells and multi-head attention for crime case and speaker type tagging [21]. Graph2Speak analyzed the criminal network topology and audio data for speaker identification using CSI and a simulated dataset [13]. For scene understanding, zero-shot classification through prompting LLM and a commonsense-based knowledge graph (i.e., ConceptNet) was performed [16,23]. An unsupervised screenplay summarization technique analyzed graphs via BiLSTM equipped with an attention mechanism [20]. Pre-training strategies for two visiolinguistic models, i.e., ViLBERT and VisualBERT, were applied [24]. Previously,

graph mining, language, vision, and sequence models were adopted to design different applications for crime drama understanding. So far, $ATOMIC^{20}_{20}$ through COMET(BART) with LLMs, sequence models, and deep multi-view learning is not used for any NLP classification task.

8 Conclusion and Future Work

RiMCR derives multiview sentence and commonsense-based features through BERT and COMET(BART). It forwards derived features through dual view specific independent stacked LSTMs with self-attention, to detect quintessential sequential patterns. Four types of features are fused and forwarded to a decoder for solving binary and multiclass prediction tasks.

Deploying RiMCR to understand other drama types related to comedy, tragedy, or melodrama is also essential. RiMCR can also be applied for sentiment analysis in different domains other than drama understanding. Model ablation of RiMCR performed through multiview graph clustering on 23 relations of $ATOMIC^{20}_{20}$ can help to identify useful features and improve RiMCR for different applications. Using multiview commonsense features in COMET(BART) or applying RiMCR to interpret real-life criminal mysteries or autopsy reports is interesting. RiMCR solely processes textual data; it would be beneficial to incorporate additional modalities such as audio or video. Adopting RiMCR to identify suspicious activities in videos recorded through surveillance cameras is profitable.

Acknowledgements This work is partially funded by the Higher Education Commission (HEC) of Pakistan to support law enforcement agencies for robust crime investigation and prevention.

Availability of data and materials The dataset and supplementary material are available here: `https://github.com/uqsameen/CommonsenseKG`

Authors' contributions Abdullah Zia: Data Labeling and source code. Abdullah Zia and Sameen Mansha: Conceptualization and methodology design. Sameen Mansha: Writing preliminary draft and debugging code. Faisal Kamiran: Supervision, investigation and validation. All authors: Revision of the manuscript.

References

1. Bart. `https://huggingface.co/docs/transformers/model_doc/bart`
2. Bert-base(uncased). `https://huggingface.co/google-bert/bert-base-uncased`
3. Comet-bart. `https://github.com/allenai/comet-atomic-2020/tree/master/models/comet_atomic2020_bart`
4. Distilbert. `https://huggingface.co/distilbert/distilbert-base-uncased`
5. Edinburghnlp. `https://github.com/EdinburghNLP/csi-corpus`

6. Foreverdreaming. http://transcripts.foreverdreaming.org
7. Huggingface transformers. https://huggingface.co/docs/transformers/index
8. Mpnet. https://huggingface.co/docs/transformers/model_doc/mpnet
9. Robertalarge. https://huggingface.co/FacebookAI/roberta-large
10. Wikipedia. https://en.wikipedia.org/wiki/CSI:_Crime_Scene_Investigation
11. AlGhamdi, M., Abdel-Mottaleb, M.: Dual-view deep convolutional neural network for matching detected masses in mammograms. Comput. Methods Programs Biomed **207**, 106152 (2021)
12. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
13. Fabien, M., Sarfjoo, S.S., Motlicek, P., Madikeri, S.: Graph2speak: Improving speaker identification using network knowledge in criminal conversational data. arXiv preprint arXiv:2006.02093 (2020)
14. Frermann, L., Cohen, S.B., Lapata, M.: Whodunnit? crime drama as a case for natural language understanding. TACL **6**, 1–15 (2018)
15. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: IEEE ICASSP. pp. 6645–6649 (2013)
16. Harrando, I., Reboud, A., Schleider, T., Ehrhart, T., Troncy, R.: Proze: Explainable and prompt-guided zero-shot text classification. IEEE IC **26**(6), 69–77 (2022)
17. Hwang, J.D., Bhagavatula, C., Bras, R.L., Da, J., Sakaguchi, K., Bosselut, A., Choi, Y.: Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs (2021), https://arxiv.org/abs/2010.05953
18. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension (2019)
19. Liu, H., Singh, P.: Conceptnet—a practical commonsense reasoning tool-kit. BT technology journal **22**(4), 211–226 (2004)
20. Papalampidi, P., Keller, F., Frermann, L., Lapata, M.: Screenplay summarization using latent narrative structure. In: EMNLP. pp. 1920–1933 (2020)
21. Papasarantopoulos, N., Frermann, L., Lapata, M., Cohen, S.B.: Partners in crime: Multi-view sequential inference for movie understanding. In: EMNLP-IJCNLP
22. Qin, Y., Zhang, X., Yu, S., Feng, G.: A survey on representation learning for multi-view data. Neural Networks p. 106842 (2024)
23. Reboud, A., Harrando, I., Lisena, P., Troncy, R.: Stories of love and violence: zero-shot interesting events' classification for unsupervised tv series summarization. Multimedia Systems **29**(6), 3951–3969 (2023)
24. Reboud, A., Troncy, R.: What you say is not what you do: Studying visio-linguistic models for tv series summarization. In: IEEE/CVF ICV. pp. 3149–3153 (2021)
25. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
26. Sap, M., Le Bras, R., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N.A., Choi, Y.: Atomic: An atlas of machine commonsense for if-then reasoning. In: AAAI. vol. 33, pp. 3027–3035 (2019)
27. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeuRIPS (2017)
29. Yan, X., Hu, S., Mao, Y., Ye, Y., Yu, H.: Deep multi-view learning methods: A review. Neurocomputing **448**, 106–129 (2021)