Modeling Human Motives and Emotions from Personal Narratives Using External Knowledge And Entity Tracking

Prashanth Vijayaraghavan
MIT Media Lab
Cambridge, Massachusetts, USA
pralav@mit.edu

Deb Roy
MIT Media Lab
Cambridge, Massachusetts, USA
dkroy@media.mit.edu

ABSTRACT

The ability to automatically understand and infer characters' motivations and emotional states is key to better narrative comprehension. In this work, we propose a Transformer-based architecture, referred to as NEMO, to model characters' motives and emotions from personal narratives. Towards this goal, we incorporate social commonsense knowledge about the mental states of people related to social events and employ dynamic state tracking of entities using an augmented memory module. Our model learns to produce contextual embeddings and explanations of characters' mental states by integrating external knowledge along with prior narrative context and mental state encodings. We leverage weakly-annotated personal narratives and knowledge data to train our model and demonstrate its effectiveness on publicly available STORYCOMMONSENSE dataset containing annotations for character mental states. Further, we show that the learned mental state embeddings can be applied in downstream tasks such as empathetic response generation.

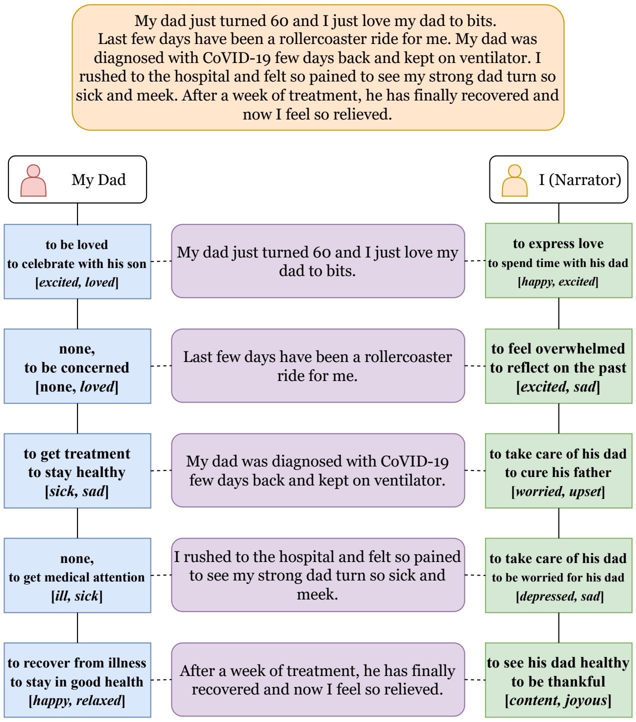
CCS CONCEPTS

• **Computing methodologies** → *Information extraction*; **Natural language generation**; *Knowledge representation and reasoning*; *Discourse, dialogue and pragmatics.*

KEYWORDS

natural language generation, narrative comprehension, representation learning, pragmatics, memory network, mental state representation, entity tracking, social events, social commonsense Knowledge, external memory

ACM Reference Format:
Prashanth Vijayaraghavan and Deb Roy. 2021. Modeling Human Motives and Emotions from Personal Narratives Using External Knowledge And Entity Tracking. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449997



My dad just turned 60 and I just love my dad to bits. Last few days have been a rollercoaster ride for me. My dad was diagnosed with CoVID-19 few days back and kept on ventilator. I rushed to the hospital and felt so pained to see my strong dad turn so sick and meek. After a week of treatment, he has finally recovered and now I feel so relieved.

My Dad		I (Narrator)
to be loved **to celebrate with his son** **[*excited, loved*]**	My dad just turned 60 and I just love my dad to bits.	**to express love** **to spend time with his dad** **[*happy, excited*]**
none, **to be concerned** **[none, *loved*]**	Last few days have been a rollercoaster ride for me.	**to feel overwhelmed** **to reflect on the past** **[*excited, sad*]**
to get treatment **to stay healthy** **[*sick, sad*]**	My dad was diagnosed with CoVID-19 few days back and kept on ventilator.	**to take care of his dad** **to cure his father** **[*worried, upset*]**
none, **to get medical attention** **[*ill, sick*]**	I rushed to the hospital and felt so pained to see my strong dad turn so sick and meek.	**to take care of his dad** **to be worried for his dad** **[*depressed, sad*]**
to recover from illness **to stay in good health** **[*happy, relaxed*]**	After a week of treatment, he has finally recovered and now I feel so relieved.	**to see his dad healthy** **to be thankful** **[*content, joyous*]**

Figure 1: Sample personal narrative is shown on the top. It contains the motives and emotional reactions [*italics*] of different characters – dad and son (narrator) in the narrative.

1 INTRODUCTION

Narratives are one of the most common yet powerful means of communication used to enhance engagement with people's issues and understanding of the social world. People share and consume them in a variety of ways to convey and make sense of their experiences. Theorists and researchers in a wide variety of fields like neuroscience, psychology, and narratology have long posited that narratives exert a powerful influence on social cognition by evoking mentalizing process [8, 9, 17, 22]. Mentalizing is used to describe all kinds of reasoning about others' mental states, such as inferring other people's thoughts, beliefs, attitudes, emotions, and motivations. Studies have argued that reading more stories in one's lifetime and analyzing characters' behavior in stories contributes to greater activation of mentalizing network [33]. Therefore, comprehending narratives is key to understanding human agency.

In this work, we are specifically interested in uncovering certain aspects of the relationship between narratives and mentalizing [14, 32, 34, 50]. We focus on developing computation approaches to model human motives and emotions from narratives containing explicit and implicit references to the characters' psychological states and their corresponding social contexts. To this end, different

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449997

models of narrative analysis such as Labov's "evaluative devices" [19], or Lehnert's "plot units" [29] have been proposed to track the mental states or affect states of the characters towards narrative understanding and summarization. A work by [41, 42] focused on constructing a dataset comprising rich low-level annotations of categories and textual explanations of motivations and emotional reactions of characters in five-sentence stories. By modeling character-specific contexts and pretraining on free-text responses, they provide benchmark results on this new resource. However, very limited work focuses on rich representation and generation of textual explanations of mental states, precisely motives and emotional reactions. Also, there is tremendous scope for improvement in furthering the research towards imparting mentalizing capabilities for machines. Some of the key challenges in modeling human motives and emotions include: (a) lack of annotated data that captures explicit and implicit mental states of characters in narratives from different domains, (b) ability to track characters' mental state shifts continuously, and (c) effectively embed and generate their corresponding text explanations.

To tackle a subset of the aforementioned challenges, we resort to personal narratives from social media. Similar to a literary story, a personal narrative is likely to contain a beginning, middle, and end, where the middle typically presents a complication for the person, one that is resolved in some way by the ending. Similarly, it may convey information about goals, motives, thoughts, conflicts, emotions, and resolutions of people, including self or other people inside or outside their social circle [1, 15]. This makes them a practical resource for knowledge extraction and modeling. Since manual annotation is usually labor-intensive and expensive, we adopt a combination of web data mining and information extraction (IE) strategies to automatically extract and aggregate noisy expressions of motivations and emotions related to specific events in the text (applicable to different textual domains). This facilitates the acquisition of weakly-annotated data containing characters' motivations and emotions from personal narratives and social commonsense knowledge from the web. Figure 1 (top) shows a sample personal narrative from Reddit with character-specific explanations of intents and emotions behind every event in the narrative. Consider the sentence "I rushed to the hospital...", the intent of the narrator ("I") is "to take care of his dad" or "to be worried for his dad". To produce such explanations, it is necessary to condition on the story context and social role because modifying them could significantly alter the meaning and their corresponding intent and emotional reactions behind the same action (e.g. "doctor rushed to the hospital" could have a different intent: "to attend to an emergency patient").

Thus, our goal is to (a) develop rich representations of mental states of humans grounded in intuitive theories of human psychology and commonsense knowledge, (b) generate textual explanations of mental states considering the prior context and social role information, and (c) harness transferability to downstream tasks. We, therefore, implement a Transformer-based encoder-decoder architecture, referred to as Nemo[1] to embed and explain characters' (or entities') mental states. To this end, we equip our model with components that: (a) enable pragmatic enrichment of narrative sentences using the aggregated knowledge and (b) track entities' mental states over time using an external memory module. Inspired by the ideas from cognitive science [21], these components can be perceived as analogous to certain characteristics of semantic and episodic memories. Thus, our contributions are as follows:

- Data collection of Personal Narratives [2] and Social Commonsense Knowledge containing weak-annotations of motivation and emotion text expressions.
- An end-to-end Transformer-based Nemo model augmented with modules that infuse social commonsense knowledge and dynamically track entities' mental states.
- Trained on the aggregated weakly annotated data, we conduct experiments on the Story- Commonsense dataset [41] under various evaluation settings. To exemplify our learned embeddings' transferability, we perform a simple evaluation on EmpatheticDialogues dataset.

2 PROBLEM SETUP

Formally, a story $\mathcal{S}$ consists of a sequence of T sentences $\mathcal{S} = [s^{(1)}, s^{(2)}, .., s^{(T)}]$ and a set of N entities/characters $\mathcal{E} = \{e_1, e_2, .., e_N\}$. We denote t^{th}-sentence containing L words as $s^{(t)} = [w_1^{(t)}, w_2^{(t)}, ..., w_L^{(t)}]$. Given an entity e_j, current story sentence $s^{(t)}$ and prior story context $s^{(<t)}$, we aim to generate mental state explanations of e_j, $\mathcal{Y}_m = [y_m^{(1)}, y_m^{(2)}, ..., y_m^{(T)}]$, related to mental state attribute $m \in \{xIntent, xReact\}$. Therefore, our approach models the conditional probability: $P(y_m^{(t)}|s^{(t)}, s^{(<t)}, y_m^{(<t)}, e_j, m)$.

3 DATASET COLLECTION

Our data collection pipeline is depicted in Figure 2. We aggregate two datasets: (a) weakly-annotated personal narratives corpus and (b) Search-based Social Commonsense Knowledge (Sb-Sck). The former is intended to capture the motives and emotions extraction considering the entire story context. These are generally explicitly mentioned by the narrator in their stories. One of the limitations of the personal narratives corpus is that it may not contain implicit mental state mappings (motives & emotions) for several events in the narratives. To alleviate this limitation, we collect sentence-level implicit mental states by adopting a combination of web data mining and information extraction strategies. We elaborate on the steps involved in our data collection process in the following sections.

3.1 Personal Narratives Corpus

We construct a corpus of personal narratives by gathering posts from Reddit related to daily interactions, life experiences, relationships, comical or embarrassing situations, to name a few. Using Pushshift API[3], we aggregate 887,441 posts from specific subreddits: /r/offmychest and /r/confessions. Of these posts, we discard all those posts with tags like "[Deleted]", "NSFW" [4] or "over_18" field set to true. The number of sentences in the posts ranges from 1 to 1015. Further, we remove texts containing less than three sentences, based on Prince's definition [37] of a minimal story as consisting of a starting state, an event, and an ending state. We compute the 90^{th}-percentile of the story lengths and remove those that exceed

[1] Short for **N**arrative **E**ntity **M**ental m**O**del

[2] We will be making the data available soon.

[3] https://pushshift.io/

[4] NSFW – not safe for work

this length. This augurs well for our specific interest in short personal narratives. Therefore, we are left with 439,408 posts, with an average length of 12.08 sentences. Figure 3b (Down) shows the data distribution related to their lengths.

To create our dataset related to motivations, we look for specific expressions associated with intents or purpose. Human motivations and emotions can be expressed linguistically in several ways, sometimes with explicit use of purpose clauses. Generally, purpose clauses take the form: To-Infinitive; (In order/So as) + To-Infinitive, (so that) + Subject + Verb; For + Noun/'ing'-form. In order to systematically identify text expressions that specify motivation, we leverage OpenIE[5] methods [2, 49] to extract a list of propositions usually composed of a single predicate and an arbitrary number of arguments. Using PropBank [35] and its annotation scheme, we can break down syntactically complex sentences as: (a) Arg-0 related to the argument exhibiting features of prototypical agent and (b) Argm-Prp related to the purpose or motivation expressions in the text. Figure 2b shows a sample OpenIE extraction of agent and its purpose. One of the authors assessed the extraction quality by analyzing a random subset of the agent-purpose pairs for each type of purpose clause and their context. By eliminating trivial extractions with a basic classifier (see Appendix A.1.1), we use the filtered data as our weakly-annotated training data. Further, we augment these extracted motivation texts with their paraphrases using a back-translation approach [11] to simulate multiple-annotation settings. A pretrained English↔German translation model is used for this purpose (e.g., to divert attention in tough times → to distract attention in difficult times).

We adopt a similar strategy to extract the emotions of characters in the narratives. First we identify 400 keywords extracted from a combination of: (a) emotion-directed[6] lexical units from FrameNet [4] corresponding to different emotions, and (b) emotion vocabulary list[7]. Though we don't have any semantic role labeling for emotions, we still feed the sentences through the OpenIE extraction method. By examining extracted propositions, we discard those story sentences when: (a) sentence is negative (contains not), (b) emotion keyword is not a part of the predicate, and (c) the first argument is neither a noun nor pronoun. Using the first argument as the agent experiencing the emotion and lexical units specified in FrameNet to express feelings footnoteWe choose semantic frames related to "Feeling" (e.g., verbs like feel, experience, get, be; phrases like sense of, feelings of, full of), we map specific sentences in the narrative to the particular character and its emotion expressions. We accomplish this by utilizing spaCy's rule-based matching tool[8] to capture particular patterns in text. The data statistics are given in Table 1. Sample extractions are highlighted in Figure 3a.

Three non-author annotators labeled a random sample of 300 instances (balanced between intent & emotions) for validation. Given the narrative context up to the sentence of interest, each annotator is asked to choose the right intent or emotion explanation expressed or implicitly felt by the character in the narrative. We let the annotators choose from the candidate texts that are: (a) extracted using our method, (b) chosen randomly, or (c) None (if the annotators feel there is no clear intent or emotion for any instance). We find that the annotators agree with our extracted intents (Fleiss' $\kappa = 0.87$) and emotion (Fleiss' $\kappa = 0.90$) texts in 89% and 93% of the cases respectively.

Personal Narratives Corpus	
#total narratives	439,408
#avg characters per story	2.02
#narratives w/ mappings	85,587
#sents w/ motives	167,256
#sents w/ emotions	318,872
% first-person motives	48.01%
% first-person emotions	58.26%
Sb-Sck Dataset	
#events w/ motives	103,357
#events w/ emotions	69,584
#unique social roles	586

Table 1: Statistics of Personal Narratives Corpus (top) and Search-based Social Commonsense Knowledge (Sb-Sck) dataset (bottom).

3.2 Social Commonsense Knowledge

The methods discussed in previous sections do not capture implicit mental states. To obtain those implicit states, we resort to: (a) exploiting social commonsense knowledge (Sck) obtained from existing sources like Atomic [45] and ConceptNet [31] and (b) mining the web to augment more knowledge about the events from personal narrative corpus. While Atomic contains inferential knowledge based on 24k short events, the knowledge from ConceptNet may not align with our requirements. For our purpose, we choose ConceptNet's relevant relations: /r/MotivatedByGoal, /r/CausesDesire, /r/Entails, /r/Causes, /r/HasSubevent.

In our work, we posit that social roles (e.g., student, mother, boyfriend, etc.) provide extra information about the motives and emotions behind an action. The base events in knowledge sources like Atomic contain typed markers (e.g., *PersonX*) where such information is lost. Therefore, we adopt web-based knowledge mining techniques to account for this extra information. The quality of such assertions may not be as high as well-curated knowledge collections like Atomic. However, they can act as a great source for pretraining our models. Therefore, we undertake the following steps to aggregate more knowledge: (a) process texts from our personal narratives corpus, (b) extract propositions from text using OpenIE tools, (c) perform a web search for plausible intents and emotions by attaching purpose clauses and feelings lexical units (explained earlier) and (d) finally, remove the poorly extracted facts using a simple classifier trained on some seed commonsense knowledge. The steps involved in data collection are explained in Appendix A.1.2. Figure 3b (Top) shows an example of how the same action could have different social role-related motivations. We refer to this as Search-based Social Commonsense Knowledge (Sb-Sck) data. The data statistics are presented in Table 1.

[5] https://demo.allennlp.org/open-information-extraction/
[6] https://framenet.icsi.berkeley.edu/fndrupal/luIndex
[7] https://www.enchantedlearning.com/wordlist/emotions. shtml#wls-id-0
[8] https://spacy.io/usage/rule-based-matching

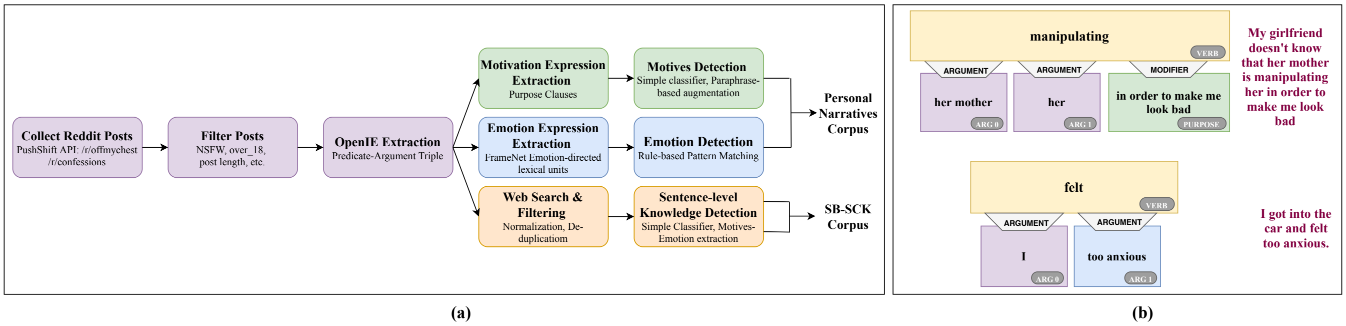
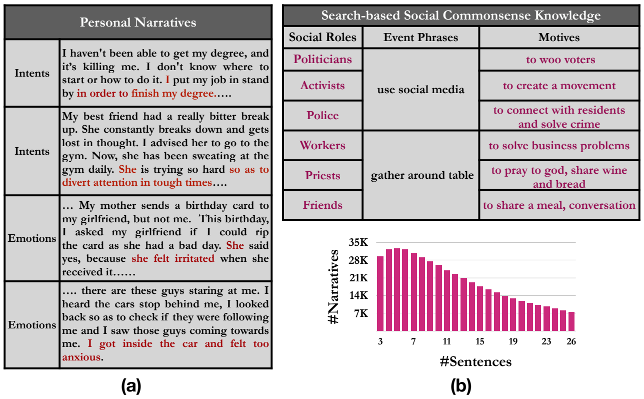

Collect Reddit Posts
PushShift API: /r/offmychest /r/confessions
Filter Posts
NSFW, over_18, post length, etc.
OpenIE Extraction
Predicate-Argument Triple
Motivation Expression Extraction
Purpose Clauses
Emotion Expression Extraction
FrameNet Emotion-directed lexical units
Web Search & Filtering
Normalization, De-duplication
Motives Detection
Simple classifier, Paraphrase-based augmentation
Emotion Detection
Rule-based Pattern Matching
Sentence-level Knowledge Detection
Simple Classifier, Motives-Emotion extraction
Personal Narratives Corpus
SB-SCK Corpus
(a)
manipulating
VERB
ARGUMENT
ARGUMENT
MODIFIER
her mother
ARG 0
her
ARG 1
in order to make me look bad
PURPOSE
My girlfriend doesn't know that her mother is manipulating her in order to make me look bad
felt
VERB
ARGUMENT
ARGUMENT
I
ARG 0
too anxious
ARG 1
I got into the car and felt too anxious.
(b)

Figure 2: Dataset Collection: (a) Illustration of the pipeline; (b) Sample OpenIE extraction for capturing motives and emotions.



Personal Narratives	
Intents	I haven't been able to get my degree, and it's killing me. I don't know where to start or how to do it. I put my job in stand by in order to finish my degree.....
Intents	My best friend had a really bitter break up. She constantly breaks down and gets lost in thought. I advised her to go to the gym. Now, she has been sweating at the gym daily. She is trying so hard so as to divert attention in tough times....
Emotions	... My mother sends a birthday card to my girlfriend, but not me. This birthday, I asked my girlfriend if I could rip the card as she had a bad day. She said yes, because she felt irritated when she received it......
Emotions	.... there are these guys staring at me. I heard the cars stop behind me, I looked back so as to check if they were following me and I saw those guys coming towards me. I got inside the car and felt too anxious.

(a)

Search-based Social Commonsense Knowledge		
Social Roles	Event Phrases	Motives
Politicians	use social media	to woo voters
Activists		to create a movement
Police		to connect with residents and solve crime
Workers	gather around table	to solve business problems
Priests		to pray to god, share wine and bread
Friends		to share a meal, conversation

(b)

Figure 3: Dataset details: (a) Samples from Personal Narratives Corpus, (b) Top: Samples from Search-based Social Commonsense Knowledge (SB-SCK) dataset, Bottom: Personal Narrative Statistics – No. of narratives w.r.t their lengths

4 RELATED WORK

There has been a growing interest in developing computational models to model aspects of human behavior from day-to-day events or stories. Prior work by [20] presented a system Aesop that builds on the idea of Lehnert's plot units [29] and utilizes existing resources to predict affect states of characters in Aesop Fables. A line of work by [10, 40] focused on modeling desire and fulfillment. This work considers five or fewer sentences to model the context of the desire expression and developed a logistic regression-based classifier for the desire fulfillment prediction task. There has been a recent body of research [18, 23] that detects emotional stimuli in stories and generates text based on specific attributes like sentiment or affect states based on LIWC categories. One of the closest works in this space is [41]'s resource for character mental state tracking in short five-sentence commonsense stories. In our work, we develop automatic techniques to extract weakly-annotated mental state expressions with social role information being retained from personal narratives (a more natural setting) and propose a method to leverage social commonsense knowledge to generate and classify character motivation and emotion states efficiently.

Further, we address the modeling challenges by incorporating social commonsense knowledge from social events and employing entity modeling for tracking the mental states of characters in the narrative.

Prior work in entity modeling is limited by their ability to track simple attributes, entity reference or specific physical properties of entities as in [6, 25, 27]. In this work, we focus on capturing the dynamics of the entities' previous motivation and emotional states. We achieve this by equipping our model using a memory module with operations involving decoder contextual hidden states. It is worth noting that models that incorporate entity-aware memory-based target-side context are a rarity. We intuit that employing attention mechanism over prior decoder states (target-side context) facilitates improved explanation generation by efficiently recording the motivation and emotion states.

5 NEMO: OUR PROPOSED MODEL

Our overall objective is to learn character-specific embeddings of mental states – especially motives and emotional reactions, and produce their textual explanations by integrating external knowledge along with social role information, preceding narrative context and mental state encodings. In this direction, we introduce a Transformer-based encoder-decoder architecture augmented with external memory modules that enable knowledge-enrichment and dynamic state tracking of entities. Figure 4 provides an overview of our NEMO model. The prime components of our model include:

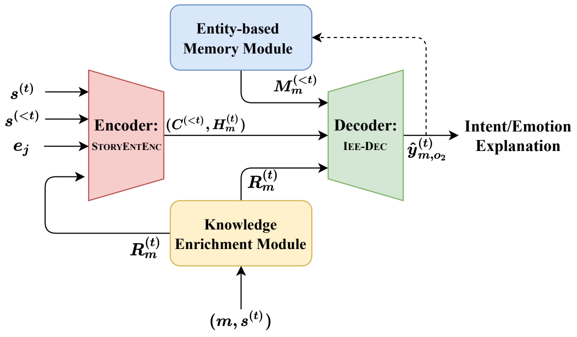


Figure 4: Overview of our NEMO model.

- *Knowledge-Enrichment Module* (Kem): Following a recent work by [52], we utilize a pretrained EventBert for this component. EventBert leverages social commonsense knowledge to sharpen the social event embeddings with semantic and pragmatic attributes. Here, the pragmatic properties refer to the human's inferred implicit understanding of event actors' intents and feelings or reactions. We feed the mental state attribute m and the current story sentence $s^{(t)}$ as our input and get a sentence-level attribute-specific pragmatics-aware embedding $R_m^{(t)}$ as the output of this module.

- *Story Entity Encoder*(StoryEntEnc): Our modified Transformer-based encoder is employed to produce prior story context embedding ($C^{(<t)}$) and entity-aware representation (H_m^t) of the current story sentence ($s^{(t)}$) consolidating the prior story sentences ($s^{(<t)}$), entity (e_j) and mental state attribute-specific pragmatics-aware knowledge embedding ($R_m^{(t)}$) obtained from (Kem).

- *Entity-based Memory Module* (Emm) : This module is used to dynamically track the prior mental states of characters in the narrative so that the generated explanations are coherent to the previous events in the narrative. Therefore, we keep track of previously generated mental state representations in a separate memory indexed using each entity (e_j) and mental state attribute information (m) and denoted as $\mathcal{M}[e_j, m]$. This module is accessed during the decoding phase by attending over memory cells to obtain attribute-specific prior mental state embeddings ($M_m^{(<t)}$).

- *Intent-Emotion Explanation Generator*(Iee-Dec): Our two pass-iterative decoder generates intent and emotion explanations by processing the encoder outputs ($C^{(<t)}, H_m^{(<t)}$), Kem output ($R_m^{(t)}$), and attribute-specific entity e_j's prior mental state embeddings $M_m^{(<t)}$retrieved from Emm.

5.1 Story Entity Encoder

Figure 5 presents a closer look into the model architecture. A variant of the conventional Transformer encoder is used to produce an entity-aware representation of the story. We introduce additional sub-layers to incorporate prior context, entity, and mental state attribute information. Our encoding strategy, StoryEntEnc(·), is defined as:

$$(C^{(<t)}, H_m^{(t)}) = \textsc{StoryEntEnc}(s^{(t)}, s^{(<t)}, e_j, m) \qquad (1)$$

where $e_j \in \mathcal{E}$ is the entity under consideration, $H_m^{(t)}$ is the resulting entity-aware representation of the story at t^{th}-step. We denote the story-specific entity embeddings as $E_{e_j} \in \mathcal{R}^{d_e}$.

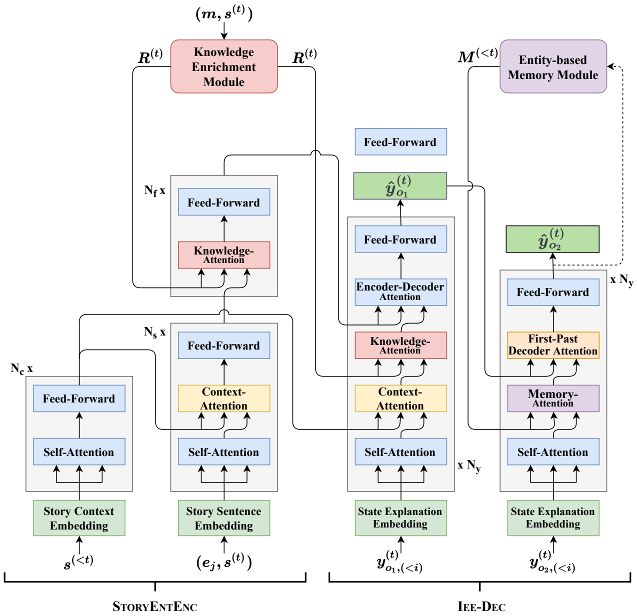


Figure 5: Illustration of the full architecture of our Nemo model.

Our StoryEntEnc is composed of a stack of N_s identical layers. To create an entity-specific understanding of the story, we perform the following steps: (a) concatenate the character information along with the current sentence to produce entity or character-aware representation of the story sentence, (b) introduce an additional context-attention sub-layer that integrates story context into the encoder, and (c) fuse knowledge representation related to specific mental state attributes. The entity concatenated input sentence is given as: $[CLS]\ e_j\ [SEP]\ w_1^{(t)}, ..., w_L^{(t)}\ [SEP]$ and $E_s^{(t)}$ is its corresponding matrix containing d_w-dimensional word-embedding vectors (in our case, $d_e = d_w$). Using steps (a) and (b), we integrate the interactions between entity-specific information from the current story sentence and its prior context. This process is given as follows:

$$U^{(l)} = \textsc{Mha}(H_s^{(l-1)}, H_s^{(l-1)}, H_s^{(l-1)}) \qquad (2)$$

$$V^{(l)} = \textsc{Mha}(U^{(l)}, C^{(<t)}, C^{(<t)}) \qquad (3)$$

$$H_s^{(l)} = \textsc{Ffl}(V^{(l)}) \qquad (4)$$

where l is the encoding layer, $l \in \{1, 2, ..., N_s\}$ and $H_s^{(0)} = E_s^{(t)}$, $C^{(<t)}$ is the prior story context embedding as computed in Section 5.1.1 and $H_s^{(l)}$ is the embedding of the source sentence at the l^{th} layer. Finally, we fuse the knowledge representation ($R_m^{(t)}$) related to specific mental state attribute (m) obtained from (Kem) with the output at N_s^{th} layer ($H_s^{(N_s)}$). The fusion step involves N_f additional Transformer layers with the context-attention replaced by knowledge-attention i.e. $\textsc{Mha}(H_s^{N_s}, R_m(t), R_m(t))$. We found in preliminary experiments that even a single fusion layer is effective in outperforming our baselines. The output from the fusion layer is the final encoded story representation, $H_m^{(t)}$, encapsulating context, entity and attribute-specific knowledge information.

5.1.1 Context-Attention & Gating. We implement standard Transformer encoder layers for computing the story context information from previous sentences $s^{(<t)}$. For the prior story context $s^{(<t)}$, we insert a $[CLS]$ and $[SEP]$ token at the start and end of each sentence, respectively. Since we add new sub-layers in this work, we

introduce a gating mechanism instead of residual connections to prevent the uncontrolled influence of information from sub-layers over the current sentence representation:

$$\beta = \sigma(W_1 H + W_2 f(H)) \tag{5}$$

$$G(H) = \beta \odot f(H) + (1 - \beta) \odot H \tag{6}$$

where f refers to the sub-layers, $\sigma(\cdot)$ is a sigmoid function, W_1, W_2 are learnable parameters.

5.2 Intent-Emotion Explanation Generator

Motivated by human cognitive behaviors, we explore the process of deliberation into the sequence generation framework [53]. This is implemented as a two pass-iterative decoding strategy. During the first pass, the decoder generates a rough draft of the explanations ($\hat{y}_{o_1}^{(t)}$) by considering sentence-level knowledge along with encoder outputs and prior context. The first step decoding outputs are fed to the second pass decoder along with entity's mental state context obtained from an entity-based memory module Emm. Formally, the two-step decoding procedure is denoted as:

$$\hat{y}_{m,o_2}^{(t)} = \text{Iee-Dec}(H_m^{(t)}, C^{(<t)}, R_m^{(t)}, M_m^{(<t)}) \tag{7}$$

where $R^{(t)}$ is the sentence-level knowledge embedding from Kem, $M_m^{(<t)}$ is the attribute-specific entity's prior mental state embeddings retrieved from Emm. In order to generate entities' intent and emotion explanations, we introduce artificial tokens associated with mental state attributes $m \in \{xIntent, xReact\}$ as the start token. For brevity, we drop the subscript m from the equations.

5.2.1 First-pass Decoding. Just like the encoder, our decoder has N_y stacked identical layers. We augment each layer with context and knowledge-attention sub-layers. While the former provides prior story context representation (extracted from $s^{(<t)}$), the latter captures the attribute-specific sentence-level knowledge information ($R^{(t)}$). The first-pass decoding procedure is explained as follows:

$$U'^{(l)} = \text{Mha}(H_{o_1}^{(l-1)}, H_{o_1}^{(l-1)}, H_{o_1}^{(l-1)}) \tag{8}$$

$$V'^{(l)} = \text{Mha}(U'^{(l)}, C^{(<t)}, C^{(<t)}) \tag{9}$$

$$W^{(l)} = \text{Mha}(V'^{(l)}, R^{(t)}, R^{(t)}) \tag{10}$$

$$Z^{(l)} = \text{Mha}(W^{(l)}, H^{(t)}, H^{(t)}) \tag{11}$$

$$H_{o_1}^{(t)} = \text{Ffl}(Z^{(l)}) \tag{12}$$

where $l \in \{1, 2, .., N_y\}$, $H_{o_1}^{(l-1)}$ is the output from previous layer, and $H_{o_1}^{(0)} = [y_0^{(t)}, y_1^{(t)}, ..., y_{i-1}^{(t)}]$ denotes the representation of words generated up until the i^{th} step ($y_{<i}^{(t)}$). Before feeding our computed representations to a feed-forward layer, we integrate the representation of the current sentence from the encoder using encoder-decoder attention. At the end of N_y layers, we compute word probabilities for the first-pass decoded sequence: $P(\hat{y}_{o_1}^{(t)}) = softmax(H_{o_1}^{(N_y)})$. Here $\hat{y}_{o_1}^{(t)}$ is the first-pass decoding output.

5.2.2 Second-Pass Decoder. During the second-pass decoder, we contextualize the current entity states' using entity's prior mental state embeddings ($M^{(<t)}$) stored in an entity-specific external memory (Emm) in combination with the first-pass decoder outputs:

$$W'^{(l)} = \text{Mha}(U''^{(l)}, M^{(<t)}, M^{(<t)}) \tag{13}$$

$$Z'^{(l)} = \text{Mha}(W'^{(l)}, \hat{H}_{o_1}^{(t)}, \hat{H}_{o_1}^{(t)}) \tag{14}$$

where $U''^{(l)}$ is the second-pass counterpart of self-attention sublayer ($U'^{(l)}$) and $\hat{H}_{o_1}^{(t)}$ is the representation of words generated during the first pass. The polished mental state explanations are computed as: $P(\hat{y}_{o_2}^{(t)}) = softmax(H_{o_2}^{(N_y)})$, where $H_{o_2}^{(l)} = \text{Ffl}(Z'^{(l)})$ is the feed-forward sub-layer output. Thus, $\hat{y}_{o_2}^{(t)}$ is the polished decoded output.

5.3 Knowledge-Enrichment Module

Knowledge-Enrichment Module (Kem) can be viewed akin to a semantic memory [38]. Generally, semantic memory refers to a long-term storehouse of general knowledge related to events, facts, and concepts. The core idea is to encode a story sentence into a pragmatics-aware embedding. The pragmatic components refer to the implied emotions and intents associated with the events in the story text. By leveraging social commonsense knowledge explained in Section 3.2, we follow a recent work of [52] and utilize the EventBert as our Kem to pretrain and effectively embed both semantic and pragmatic aspects of social events.

The input is a concatenation of mental state attribute $m \in \{xIntent, xReact\}$ with the story sentence $s^{(t)}$. This is fed through the EventBert to produce attribute-specific contextualized social event embeddings, $R_m^{(t)}$. This encoding step is followed by an attentive pooling function that attends over contextual embeddings to output a summarized pragmatics-aware embedding $r_m \in \mathcal{R}^{d_h}$ reflecting intents ($m = xIntent$) and emotional reactions ($m = xReact$). We learn these representations by pretraining using an N-pair loss (as in [52]) for each intent or emotion explanations. By training on data from social commonsense knowledge sources, we enable the model to learn pragmatics-aware representation of the social events. While the contextual vectors $R^{(t)}$ are used during encoding and decoding phases, the summarized vectors $r_m^{(t)}$ are used to initialize our entity-based memory module (EMM). More details about this component are presented in Appendix C.

5.4 Entity-based Memory Module

Entity-based Memory Module can be seen as an episodic memory that ideally stores the mental states of characters in a specific narrative. To track entity-specific mental state representations, we utilize a memory, $\mathcal{M}$, containing separate memory cells for each entity e_j and mental state attribute m. Therefore, memory is indexed using entity embeddings (E_{e_j}) and mental state attribute embeddings (E_m)). For simplicity, we denote it as: $\mathcal{M}[e_j, m]$. The memory operations are explained as follows: $\mathcal{M} = (\mathcal{K}, \mathcal{A}, \mathcal{V})$, where key $\mathcal{K}$ is tied with entity embeddings, $\mathcal{A}$ refers to the mental state attribute m and $\mathcal{V}$ contains the attribute-specific target-size context vectors.

Memory Attention: Our decoder applies a multi-head attention mechanism over prior mental state representations of an entity $M_m^{(<t)}$ for each mental state attribute m. For a specific entity e_j and mental state attribute m, we retrieve $(t-1)$ memory cells from $\mathcal{M}[e_j, m]$ by masking the future time steps. Finally, we inject

the sequence-order information using positional encoding [51] to get $M^{(<t)}$ (drop the subscript m to be consistent with previous notations).

Memory Write: We keep track of prior mental states by storing their representations in our memory $\mathcal{M}[e_j, m]$. It is possible to limit the memory allocated to each entity for prior context (say n-previous sentences). However, we don't set such limits in this work. We initialize the memory with sentence-level pragmatics-aware summarized vector, $r_m^{(t)}$. For the write operation, we apply a gating mechanism to store the final decoder hidden state of $\hat{y}_{o_2}^{(t)}$ given as $h_{\hat{y}_{o_2}}^{(t,L)}$ at the t^{th} memory cell:

$$\gamma = \sigma(W_r r_m^{(t)} + W_h h_{\hat{y}_{o_2}}^{(t,L)}) \tag{15}$$

$$\mathcal{M}[e_j, m, t] = \gamma \odot r_m^{(t)} + (1 - \gamma) \odot h_{\hat{y}_{o_2}}^{(t,L)} \tag{16}$$

where W_r and W_h are learnable parameters. In our experiments, we find that this method is simple yet effective.

6 TRAINING & HYPERPARAMETERS

Our aggregated data is split into train, validation, and test sets at 70-10-20 split. Following [54]'s work, our model is trained to minimize the negative log-likelihood of predicting each word during both the decoding steps: $\mathcal{L} = \mathcal{L}_{mle1} + \mathcal{L}_{mle2}$. To handle our weakly-annotated data, we perform phase-wise training of our model. We pretrain our model using all the social commonsense knowledge data where the entity or character information is concatenated with the input text during the first phase. The memory cells are initialized to zero, and the model learns to produce sentence-level explanations. The second phase involves modeling the current story sentence along with the prior narrative context. We initialize the memory with pretrained sentence-level knowledge embedding $r_m^{(t)}$ once for a mini-batch and further update them with noisy explanations. This exposes the model to its potential test-time errors and guides the model to learn robust parameters.

Using grid-search, we tune the hyperparameters and the best configuration ($N_c = 2, N_f = 1, N_s = 12, d_h = 768$ and 12 attention heads) is obtained based on validation set perplexity. To prevent overfitting, we use dropout with a rate of 0.2. By default, we experiment with GloVe vectors and ELMo-based contextualized embeddings (usually mentioned during evaluation). We use Adam as our optimizer with a learning rate of $\alpha = 0.0002$ [28] and a training batch size of 8. We use greedy decoding at training time, but utilize beam-search with a beam size of $k = \{3, 5, 10, 12\}$ [3, 47] at inference time.

7 EXPERIMENTS

In this section, we describe the various evaluation settings: datasets, baselines, model variants, modes, and metrics. We designed our experiments to study the following research questions:

RQ1: How well does our model perform compared to other baselines in the explanation generation task? How much does each component impact the overall performance?

RQ2: Can our model representations be used to perform state classification based on labeled motivation and emotional reaction categories?

Dataset	#Stories	#Motives	#Emotions
Personal Narratives	300	882	1418
StoryCommonsense	2500	6831	13785

Table 2: Test set statistics for explanation generation task: This includes number of annotated stories and number of character-lines with motives and emotions.

RQ3: Do the learned mental state representations exhibit transfer capability to a downstream task?

7.1 Explanation Generation Task (RQ1)

7.1.1 Dataset. We run experiments on (a) the manually annotated gold explanations for sampled data from personal narratives corpus and (b) the benchmark character psychology dataset – StoryCommonsense [41]. Table 2 summarizes the dataset used for evaluation of our explanation generation task.

7.1.2 Baselines. We compare our model's performance to different baseline methods. We follow a model architecture for the baseline methods as in [41], where they compute an encoded vector by concatenating the current sentence representation along with the entity-specific context (involving sentences where a particular entity appears). These methods are enlisted as follows:

- **LSTM** [48], which is a hierarchical RNN-based encoder-decoder model. The sentence tokens are encoded using a bi-LSTM. The entity-specific vector, computed using a similar method, is then concatenated with the sentence vector.
- **Ren** [25], which is a recurrent entity network updating entity states in a dynamic long-term memory. A memory cell is initialized for every entity in the story and updated after reading every sentence. The memory vector in the cell corresponding to the entity under consideration is the final encoded vector.
- **Npn** [6], which performs dynamic entity tracking by explicitly modeling actions as state transformers. Memory is initialized and accessed as in Ren.
- **Gpt** [39], which is a fine-tuned transformer-based language model architecture. The input setup consists of the concatenation of entity marker, story context tokens, current sentence tokens, and mental state attribute token m separated by special [SEP] tokens. This is closely related to how Gpt is used in [7].

7.1.3 Model Variants. By evaluating on the personal narrative corpus, we assess the impact of three of our model components: Kem (semantic memory), Emm (episodic memory), Iee-Dec (deliberation decoder). To comprehensively study their impact, we remove them one at a time as model variants and evaluate their impact on the performance in the explanation generation task.

7.1.4 Metrics. Due to the short sequence length of generated explanations and the high possibility of producing similar explanations in multiple ways, we avoid word-overlap based metrics and instead compute embedding-based metrics such as embedding average and vector extrema for evaluating explanation generation quality [24, 30]. Embedding average calculates sentence-level embeddings

(a) Personal Narrative Corpus

Models	Motivation		Emotion	
	Avg	VE	Avg	VE
Hred	58.43	48.78	52.05	51.18
Ren	58.96	49.87	53.59	52.14
Npn	59.03	50.02	52.63	51.76
Gpt	63.56	54.77	56.74	56.09
Nemo	**69.27**	**59.78**	**62.88**	**61.34**
Comet repl.	66.55	58.23	60.78	60.21
w/o Emm	67.16	57.64	59.74	58.38
w/o Kem	65.38	56.86	60.58	60.06
w/o Iee-Dec	66.92	58.17	60.42	59.83

(b) StoryCommonsense dataset

Models	Motivation		Emotion	
	Avg	VE	Avg	VE
Random	56.02	45.75	40.23	39.98
Lstm	58.48	51.07	52.47	52.30
Ren	58.83	51.79	53.95	53.79
Npn	57.77	51.77	54.02	53.85
Gpt	60.19	52.95	55.68	55.47
Nemo	**66.25**	**59.16**	**62.78**	**61.92**

Table 3: Automatic evaluation results on (a) Personal Narratives corpus & (b) StoryCommonsense dataset. Bold face indicates leading results for the corresponding metric.

by averaging the word embeddings of each token in a sentence. Vector extrema metric takes the most extreme value for each dimension amongst all word vectors in the sentence and uses that value in the sentence-level embedding. To compare the ground truth and generated explanation, we compute the cosine similarity between their respective sentence-level vectors. These metrics Additionally, these metrics are useful for comparison with the previous benchmark used for the generation task [41].

7.1.5 Results. The main results of our evaluation on Personal narratives and StoryCommonsense datasets are summarized in Table 3a and Figure 3b respectively. We observe that our complete model achieves an absolute mean improvement of ∼ 9% and ∼ 12% over a fine-tuned GPT model using the embedding average metric of the generated intent and emotion explanations respectively across both the datasets.

Effect of architectural choices. By training variants of our Nemo model with and without specific components of the model, we are able to ascertain their importance for the task at hand. Table 3a shows that our Kem and Emm yield significant boost to the overall performance. The dip in performance on intent generation is more pronounced when Kem is removed while Emm is critical for the improved performance of emotion generation. We intuit the reason to be the additional sentence-level commonsense knowledge infused by Kem leading to better generations of intents while entities' prior states from Emm guiding the overall prediction of the current emotional state.

Effect of Knowledge Embeddings. From Table 3a, it is clear that Kem provides really good performance gains. Further, we replace the knowledge embeddings obtained from Kem with the embeddings extracted from Comet [7]. Comet is a framework that adapts the language model weights to produce diverse commonsense knowledge tuples. The scores reported in Table 3a indicate that there is a significant advantage of using Kem embeddings over Comet. Also, we note that Comet only provides a small marginal improvement in comparison to a Nemo model without Kem component. But the addition of our Kem component provides a huge jump in performance, specifically while generating motives. This can be attributed to the social role information, a characteristic of our social commonsense knowledge resource, utilized by our Kem module. We verify this in the error analysis (see Section 7.2.5).

7.1.6 Human Evaluation of Trajectories. We conduct a human evaluation to test the effectiveness of our Nemo model in generating motivation and emotion explanation trajectories. Our experiment compares our model explanations to those obtained from Gpt-based model. We randomly select 100 stories and present the story, character, and the visualization of trajectories to three workers. The workers then select the trajectory that best matches the characters' mental states. The inter-annotator agreement had a Fleiss' $\kappa = 0.74$ and $\kappa = 0.78$ for intent and emotion trajectories respectively, indicating substantial agreement among the workers. Moreover, 47 intent and 56 emotion trajectories had a unanimous agreement among three workers, of which 45 intent and 52 emotion trajectories were in favor of trajectories generated by Nemo. Based on the majority agreement, the workers selected our intent and emotion trajectories for 81% and 83% of the presented stories, respectively. Thus, it is clear that our model is able to generate better explanation trajectories.

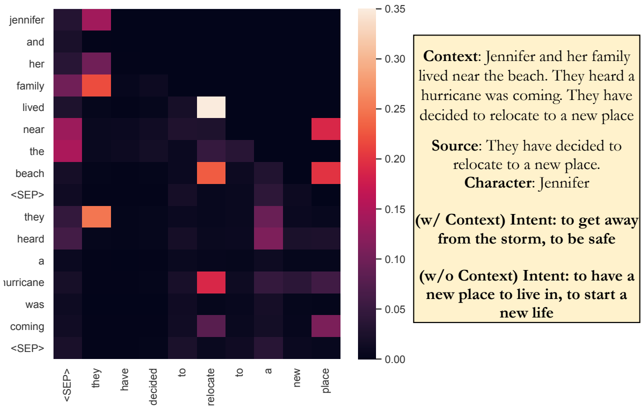
7.1.7 Qualitative Analysis.

Effect of Context. We investigate the effect of context in producing convincing explanations for our text by filtering null attention and plotting an attention map between context and source text (see Figure 6). Notably, this particular attention head (head-6) maps specific source words to their relevant context words. The attention head's focuses on the following words: "relocate" ↦ {"lived","beach", "hurricane"} and "they" ↦ {"jennifer","her", "family"}. Further, we also show sample generation with and without the context information. It is evident from these examples that Nemo can identify particular aspects of the context that are relevant (e.g., antecedents, spatial concepts) and leverage them to produce appropriate explanations.

Effect of Two-pass decoding Step. Figure 7 provides sample motivations generated by our proposed model in multiple passes along with GPT (as it performs competitively for our task). We demonstrate our model's ability to generate explanations from the narrator's perspective (1^{st} person) and that of another entity/character in the narrative. The result also shows improvement after the second-pass decoding.

7.2 State Classification Task (RQ2)

7.2.1 Dataset. The StoryCommonsense dataset comprises over 300k low-level annotations for motivations and emotions across 15,000 short stories selected from ROCStories training set [41]. This



Context: Jennifer and her family lived near the beach. They heard a hurricane was coming. They have decided to relocate to a new place

Source: They have decided to relocate to a new place.
Character: Jennifer

(w/ Context) Intent: to get away from the storm, to be safe

(w/o Context) Intent: to have a new place to live in, to start a new life

Figure 6: Attention map (head-6) between context and source. On the x-axis are the source tokens, on the y-axis the context tokens.

Input Text	**Context:** I loved Mary intensely. But she wanted to be only friends with me. **Source:** She found a guy called John	**Context**: I was joining a grad school at time when ethnic problems were rife. **Source**: As I was crossing the road to get to my class, there was a girl who was resisting the police brutality, leading from front and later I joined it too.
Models	**Entity/Character: Mary**	**Entity/Character: I (myself)**
GPT	• none • to be loved • to be happy	• to get to the class • to gain knowledge • to be safe
Our Model (First pass)	• to have a relationship • to be with him • to be loved by someone	• to get to the class quickly • to get to the class in problems • to defend someone
Our Model (Second Pass)	• to have a relationship **with john** • to **be friends with john** • to **have a relationship with the guy**	• to get to the class **on time** • to **stand up for a cause** • to **defend someone in a situation**

Figure 7: Generation of motivation explanations in multiple decoding steps.

dataset includes the categorization of motivations and emotional reactions based on different classical theories of psychology.

7.2.2 Experimental Settings & Baselines. We conduct experiments under the following settings:

- **Zero-shot (ZS)** In this setting, we map the generated emotion explanation to one of the 8 Plutchik's categories via nearest neighbor search in the word-embedding space: $\bar{y}_c = argmax_{c \in C}(cos(E_{\hat{y}_{xReact}}, E_c))$, where $c \in C$ is the label related to Plutchik's categories. Without any further fine-tuning, we compare our results against Comet-Cga [5] and use their word formulation setup for labels.
- **Supervised (SS)** We fine-tune our trained model using a feed-forward layer on the top of the encoder output. Additionally, we experiment with (Nemo$_E$) and without (Nemo$_{NE}$) annotated explanation training. In addition to the baselines in the original work, we compare against – BiLSTM + Self-Attention (**Bm**) and BiLSTM + Self-Attention + Knowledge (**Bm+K**) which incorporate multihop knowledge paths using graph-based algorithms [36] for predicting human needs (motivation categories). Additionally, we report scores from a recent work [16] that uses label semantics (referred to as Ls) and track label-label correlation for emotion inference task.
- **Low-resource regimes (LR)** This scenario has a significant practical interest, specifically in adapting our model to domains with a small amount of in-domain labeled data. Having trained on personal narratives corpus, we simulate low resource regimes by varying the percentage of training examples from StoryCommonsense state classification dataset.

7.2.3 Metrics. Consistent with prior study [41], we compute the micro-averaged F_1 scores for the state classification task: Maslow, Reiss and Plutchik states.

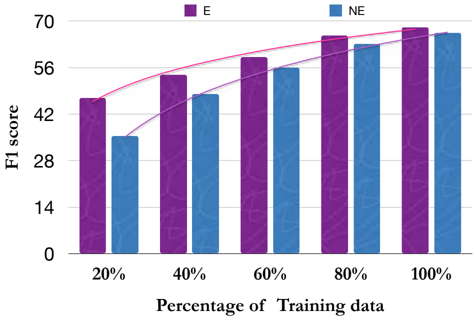


Figure 8: Prediction performance under low-resource (LR) settings (limited amounts of training data).

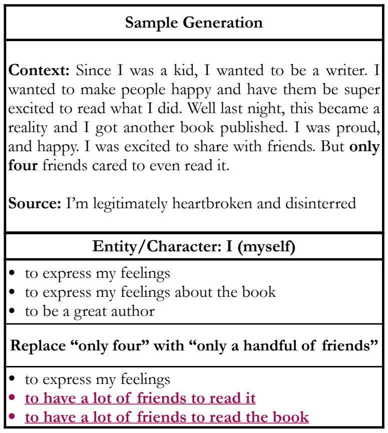
7.2.4 Results. Visibly, our models (in Table 4) outperform the state-of-the-art methods significantly in both settings. With ELMo-based embeddings, the improvements are even more pronounced. For the zero-shot settings, we report the scores directly from the original work [5]. We find a similar pattern in the state classification task for the supervised setting as in Section 7.1.5. The impact of Comet is only marginally felt while the Kem component provides a relatively huge performance boost. Interestingly, our results in Figure 8 suggest that the model variant fine-tuned with explanations learns faster with lesser in-domain labeled data than its counterpart without explanation fine-tuning. We note that both these models outperform several baselines with less than 40% of training examples. We believe that the explanation fine-tuning further sharpens the learned mental state representations as the annotations are much cleaner than our aggregated personal narratives corpus.

7.2.5 Error Analysis.

Decoding Phase. During the beam-decoding step for sentences irrelevant for a particular entity, "none" can only be predicted once, which causes other candidates in the beam to be incorrect if "none" is the appropriate answer. However, we posit that the embeddings hold richer information than the explanations generated due to the limitations in the way we implement the decoding. For the explanations, we observe that the models miss out on quantities that are expressed as numbers or ambiguous phrases used in social media personal narratives. Figure 9 shows an example where a small replacement to the input produces better explanations.

Models	Maslow	Plutchik	Reiss
Comet-Cga (ZS)	_	19.30	_
Comet-Cga (T)	_	27.50	_
Nemo (ZS)	_	42.61	_
Lstm	34.55	28.81	24.51
Cnn	35.23	30.04	24.21
Ren	33.57	30.15	20.53
Npn	31.69	30.29	17.75
Bm	53.54	_	26.57
Bm+K	56.69	_	32.96
Bm (ELMo)	59.81	_	35.49
Bm+K (ELMo)	61.72	_	36.70
Ls	_	65.88	_
Nemo$_E$	69.77	68.16	45.76
Nemo$_{NE}$	67.37	66.57	46.21
Nemo$_E$ (ELMo)	**72.09**	**71.26**	**48.52**
w/o Emm	69.13	67.19	45.50
w/o Kem	66.28	68.61	43.92
Comet repl.	66.95	69.14	43.92

Table 4: State classification performance under supervised settings. ZS: Zero-Shot Settings, T-Tuned Hyperparameters as reported in [5]



Sample Generation

Context: Since I was a kid, I wanted to be a writer. I wanted to make people happy and have them be super excited to read what I did. Well last night, this became a reality and I got another book published. I was proud, and happy. I was excited to share with friends. But **only four** friends cared to even read it.

Source: I'm legitimately heartbroken and disinterred

Entity/Character: I (myself)

- to express my feelings
- to express my feelings about the book
- to be a great author

Replace "only four" with "only a handful of friends"

- to express my feelings
- to have a lot of friends to read it
- to have a lot of friends to read the book

Figure 9: Sample generations showcasing the limitations of Nemo.

Emotion State Classification. A noticeable trend in the categorization task is the high level of cross-predictions among related emotions. Several misclassifications occur between joy-surprise and anger-disgust categories. The subtle difference between those emotion pairs makes it harder for the models to distinguish them in some cases clearly.

Intent State Classification. Since we observed only a marginal improvement with the addition of the Comet module, we compare the difference in errors made by Comet replaced Nemo model in comparison to our complete Nemo model for predicting Maslow's motivation categories. We gauge that Comet replaced model made more errors (in ~ 24.5% of the cases) when the stories contain more than one social role information. This validates our claim that the social role information captured by our Nemo with Kem module is beneficial for both the classification and generation task.

Models	PPL	AVG BLEU
Fine-Tuned	21.24	6.27
Fine-Tuned Large	**16.55**	**8.06**
EmoPrepend-1	24.30	4.36
TopicPrepend-1	25.40	4.17
Ensem-DM	19.05	6.83
Ensem-SCS+	17.06	7.64

Table 5: Automatic evaluation metrics on ED test set. Ensem-SCS+: model incorporating our learned embeddings.

7.3 Application: Empathetic Dialogue Generation (RQ3)

Natural social interactions require humans to recognize and infer others' implied emotions and respond appropriately by acknowledging their underlying feelings. Since Nemo infers motivations and emotion states from stories, we posit that the embeddings learned from such a model can lead to improved performance on this dialogue generation task.

7.3.1 Dataset. We use EmpatheticDialogues (ED) dataset, introduced by [44], for evaluating the ability of Nemo representations to improve generation of empathetic responses. The dataset consists of 25k personal dialogues grounded in specific emotional situations where a speaker was feeling a given emotion, with a listener responding. The train/ val/ test split was 19533/ 2770/ 2547 conversations, respectively.

7.3.2 Model & Baselines. Following Rashkin et al.'s prior work [43], we experiment with the ensemble of encoders that augments the encoders to incorporate the embeddings extracted from pretrained architectures. The ensemble model that incorporates our mental state representations is referred to as "Ensem-SCS+". We compare our ensemble model with other well-performing benchmarks reported in [43] involving pretrained external predictors:

- **Ensem-DM**: An ensemble model with supervision from trained Deep-moji system [12].
- **EmoPrepend-1**: Add an emotion label to the beginning of the token sequence as encoder input. This is obtained from a separate classifier that predicts emotion labels from the description of the situation.
- **TopicPrepend-1**: Similarly, the top predicted label from the supervised topic classifier is merely prepended to the beginning of the token sequence as encoder input.

7.3.3 Results. For the above baselines, we report the values directly from the original work. Table 5 shows that our Ensem-SCS+ model produces significant improvement in automated metrics, quantifying the impact of using our learned representations. Our model with a relatively lower number of parameters is able to perform closer to the best performing large model.

8 CONCLUSION

In this paper, we present a Transformer-based method to model the mental states of characters related to the events in the personal narratives. Using data mining and information extraction techniques, we aggregate weakly-annotated data to train our model known as Nemo. We show that the proposed method is able to outperform several baselines in mental state tracking task. We also observe that the pretraining on weakly-annotated data helps in improving the overall performance under low-resource settings. We believe that further improvements can be achieved in explanation generation and state categorization in cases where the text contains character-irrelevant content or non-events by introducing specialized knowledge in our model. Our analysis also demonstrated the transferability of our learned representation in a downstream empathetic response generation task. Future work could investigate the applicability of these mental state representations in modeling vital elements of narrative structures.

REFERENCES

[1] H Porter Abbott. 2020. *The Cambridge introduction to narrative.* Cambridge University Press.

[2] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).* 344–354.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[4] Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1.* 86–90.

[5] Antoine Bosselut and Yejin Choi. 2019. Dynamic knowledge graph construction for zero-shot commonsense question answering. *arXiv preprint arXiv:1911.03876* (2019).

[6] Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2017. Simulating action dynamics with neural process networks. *arXiv preprint arXiv:1711.05313* (2017).

[7] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317* (2019).

[8] Christina R Carnahan, Pamela S Williamson, and Jennifer Christman. 2011. Linking cognition and literacy in students with autism spectrum disorder. *Teaching Exceptional Children* 43, 6 (2011), 54–62.

[9] Tony Charman and Yael Shmueli-Goetz. 1998. The relationship between theory of mind, language and narrative discourse: an experimental study. *Cahiers de Psychologie Cognitive/Current Psychology of Cognition* (1998).

[10] Snigdha Chaturvedi, Dan Goldwasser, and Hal Daume III. 2016. Ask, and shall you receive? understanding desire fulfillment in natural language text. In *Thirtieth AAAI Conference on Artificial Intelligence.*

[11] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381* (2018).

[12] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524* (2017).

[13] Joshua Feldman, Joe Davison, and Alexander M Rush. 2019. Commonsense knowledge mining from pretrained models. *arXiv preprint arXiv:1909.00505* (2019).

[14] Evelyn C Ferstl, Jane Neumann, Carsten Bogler, and D Yves Von Cramon. 2008. The extended language network: a meta-analysis of neuroimaging studies on text comprehension. *Human brain mapping* 29, 5 (2008), 581–593.

[15] Vittorio Gallese and Hannah Wojciehowski. 2011. How stories make us feel: Toward an embodied narratology. *California Italian Studies* 2, 1 (2011).

[16] Radhika Gaonkar, Heeyoung Kwon, Mohaddeseh Bastan, Niranjan Balasubramanian, and Nathanael Chambers. 2020. Modeling label semantics for predicting emotional reactions. *arXiv preprint arXiv:2006.05489* (2020).

[17] Rosa M Garc'ıa-P'erez, R Peter Hobson, and Anthony Lee. 2008. Narrative role-taking in autism. *Journal of Autism and Developmental Disorders* 38, 1 (2008), 156–168.

[18] Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2017. Affect-lm: A neural language model for customizable affective text generation. *arXiv preprint arXiv:1704.06851* (2017).

[19] Amit Goyal, Ellen Riloff, and Hal Daumé III. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing.* 77–86.

[20] Amit Goyal, Ellen Riloff, and Hal Daumé III. 2013. A computational model for plot units. *Computational Intelligence* 29, 3 (2013), 466–488.

[21] Daniel L Greenberg and Mieke Verfaellie. 2010. Interdependence of episodic and semantic memory: evidence from neuropsychology. *Journal of the International Neuropsychological Society: JINS* 16, 5 (2010), 748.

[22] Nicole R Guajardo and Anne C Watson. 2002. Narrative discourse and theory of mind development. *The Journal of Genetic Psychology* 163, 3 (2002), 305–325.

[23] Lin Gui, Jiannan Hu, Yulan He, Ruifeng Xu, Qin Lu, and Jiachen Du. 2017. A question answering approach to emotion cause extraction. *arXiv preprint arXiv:1708.05482* (2017).

[24] Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2018. Towards automated customer support. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications.* Springer, 48–59.

[25] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2016. Tracking the world state with recurrent entity networks. *arXiv preprint arXiv:1612.03969* (2016).

[26] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems.* 1693–1701.

[27] Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A Smith. 2017. Dynamic entity representations in neural language models. *arXiv preprint arXiv:1708.00781* (2017).

[28] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[29] Wendy G Lehnert. 1981. Plot units and narrative summarization. *Cognitive science* 5, 4 (1981), 293–331.

[30] Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023* (2016).

[31] Hugo Liu and Push Singh. 2004. ConceptNet—a practical commonsense reasoning tool-kit. *BT technology journal* 22, 4 (2004), 211–226.

[32] Raymond A Mar. 2011. The neural bases of social cognition and story comprehension. *Annual review of psychology* 62 (2011), 103–134.

[33] Raymond A Mar. 2018. Evaluating whether stories can promote social cognition: Introducing the Social Processes and Content Entrained by Narrative (SPaCEN) framework. *Discourse Processes* 55, 5-6 (2018), 454–479.

[34] Micah L Mumper and Richard J Gerrig. 2017. Leisure reading and social cognition: A meta-analysis. *Psychology of Aesthetics, Creativity, and the Arts* 11, 1 (2017), 109.

[35] Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics* 31, 1 (2005), 71–106.

[36] Debjit Paul and Anette Frank. 2019. Ranking and selecting multi-hop knowledge paths to better predict human needs. *arXiv preprint arXiv:1904.00676* (2019).

[37] Gerald Prince. 2012. *A grammar of stories: An introduction.* Vol. 13. Walter de Gruyter.

[38] M Ross Quillan. 1966. *Semantic memory.* Technical Report. BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA.

[39] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

[40] Elahe Rahimtoroghi, Jiaqi Wu, Ruimin Wang, Pranav Anand, and Marilyn A Walker. 2017. Modelling protagonist goals and desires in first-person narrative. *arXiv preprint arXiv:1708.09040* (2017).

[41] Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018. Modeling naive psychology of characters in simple commonsense stories. *arXiv preprint arXiv:1805.06533* (2018).

[42] Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A Smith, and Yejin Choi. 2018. Event2mind: Commonsense inference on events, intents, and reactions. *arXiv preprint arXiv:1805.06939* (2018).

[43] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2018. I know the feeling: Learning to converse with empathy. (2018).

[44] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2018. Towards empathetic open-domain conversation models: A new benchmark and dataset. *arXiv preprint arXiv:1811.00207* (2018).

[45] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence,* Vol. 33. 3027–3035.

[46] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. SocialIQA: Commonsense Reasoning about Social Interactions. *arXiv preprint arXiv:1904.09728* (2019).

[47] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709* (2015).

[48] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence.*

[49] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).* 885–895.

[50] Diana I Tamir, Andrew B Bricker, David Dodell-Feder, and Jason P Mitchell. 2016. Reading fiction and reading minds: The role of simulation in the default network. *Social cognitive and affective neuroscience* 11, 2 (2016), 215–224.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems.* 5998–6008.

[52] Prashanth Vijayaraghavan and Deb Roy. 2021. Lifelong Knowledge-Enriched Social Event Representation Learning. In *16th European Chapter of the Association for Computational Linguistics (EACL 2021).*

[53] Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems.* 1784–1794.

[54] Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Modeling coherence for discourse neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7338–7345.

A APPENDIX A

A.1 Data Collection

A.1.1 Personal Narratives Corpus. We manually identify a set of 300 extracted purpose clause texts if they truly reflect the motivation behind an action. To filter trivial motivation expressions (e.g.,"to do it"), a logistic regression classifier is trained by constructing handcrafted features from text like mean word embeddings, POS tags, number of words, presence of stopwords, and entities. Eventually, we shortlist those expressions above a threshold score, $\rho_{pn} >= 0.4$.

A.1.2 Sb-Sck Dataset. We feed sentences from personal narratives to OpenIE tools which yield subject-relation-object triples. Next, we form a web search query q after normalizing [9] the triples and concatenating them with purpose clauses (for motivations) or feeling lexical units (for emotions). The query q is issued to the search engine, using its public API and enabling the spelling correction feature.

We train a simple logistic regression using manually annotated seed sets of search results to verify if they are valid candidates for knowledge extraction. We use features like average word embedding, number of words matched, exact match or approximate match, presence/number of stop words in mental state text, type of clause (purpose/feelings) and presence of entities. We use a N-V-OW representation scheme for words similar to [13], where each word is categorized into: HeadNoun, FirstVerb, and OtherWords. Finally, we discard all results below a threshold score $\rho_{sck} < 0.35$.

B INPUT PROCESSING

Inspired by [26], we identify character names in a story using Coref systems and replace them with abstract markers to prevent degenerate solutions. We do not replace words related to social roles. We randomly permute these entity markers from a set of generic markers reused across multiple stories to primarily distinguish them from other entities in the story during our training and testing process. This allows us to embed unseen entities in new stories. We denote the story-specific entity embeddings as $E_{e_j} \in \mathcal{R}^{d_e}$.

Models	Dev	Test
w/o Social Event Embeddings		
GPT2	63.3	63.0
BERT-base	63.3	63.1
BERT-large	**66.0**	**64.5**
w/ Social Event Embeddings (Kem)		
BERT-base	65.1	64.0
BERT-large	**68.7**	**67.9**

Table 6: Accuracy scores (%) of various models including event embedding (Kem) enhanced model.

C KNOWLEDGE-ENRICHMENT MODULE

The input to this module is: $x = [CLS]\ m\ [SEP]\ w_1^{(t)}, .., w_L^{(t)}\ [SEP]$. We fine-tune a Bert-based model and an attention layer with context vectors c_m to obtain embeddings $r_m \in \mathcal{R}^{d_h}$ related to mental state attributes m. Finally, we compute a combined semantic and pragmatic event embedding r_c using a projection layer. Using the aggregated social commonsense knowledge, we compute N-pair loss for r_m and r_C using representations of ground truth mental state explanations and event paraphrases, respectively. We experimented with $N = 4, 8, 16, 32$ and select the best-performing model on the validation set. For our work, we choose $N = 8$ for the pretraining.

We determine the quality of our latent social event representations by evaluating on a social commonsense reasoning benchmark – SocialIQA dataset [46]. Given a context, a question, and three candidate answers, the goal is to select the right answer among the candidates. Following [46], the context, question, and candidate answer are concatenated using separator tokens. In this work, we fine-tune the Bert-base model by feeding the latent embeddings: $r_{xIntent}, r_{xReact}$ and r_C. While the original work computed a score l using the hidden state of $[CLS]$ token, we introduce a minor modification as:

$$l = W_5\ tanh(W_1\ h_{CLS} + W_2\ r_{xIntent} + W_3\ r_{xReact} + W_4\ r_C)$$

. In this work, we only fine-tune the BERT-base model with event embeddings denoted as "BERT-base + Kem". Experimental results in Table 6 confirm that a simple enhancement with pragmatic-enriched social event embeddings can lead to improved reasoning capabilities.

[9] For example, "clean a bedroom floor" is changed as "clean bedroom floor" using weak normalization and "clean floor" under strong normalization settings.