Modeling Naive Psychology of Characters in Simple Commonsense Stories

Hannah Rashkin[†], **Antoine Bosselut**[†], **Maarten Sap**[†], **Kevin Knight**[‡] and **Yejin Choi**[†§]
[†]Paul G. Allen School of Computer Science & Engineering, University of Washington
[§]Allen Institute for Artificial Intelligence
{hrashkin,msap,antoineb,yejin}@cs.washington.edu
[‡] Information Sciences Institute & Computer Science, University of Southern California
knight@isi.edu

Abstract

Understanding a narrative requires reading between the lines and reasoning about the unspoken but obvious implications about events and people's mental states — a capability that is trivial for humans but remarkably hard for machines. To facilitate research addressing this challenge, we introduce a new annotation framework to explain naive psychology of story characters as fully-specified chains of mental states with respect to *motivations* and *emotional reactions*. Our work presents a new large-scale dataset with rich low-level annotations and establishes baseline performance on several new tasks, suggesting avenues for future research.

1 Introduction

Understanding a story requires reasoning about the causal links between the events in the story and the mental states of the characters, even when those relationships are not explicitly stated. As shown by the commonsense story cloze shared task (Mostafazadeh et al., 2017), this reasoning is remarkably hard for both statistical and neural machine readers – despite being trivial for humans. This stark performance gap between humans and machines is not surprising as most powerful language models have been designed to effectively learn local fluency patterns. Consequently, they generally lack the ability to abstract away from surface patterns in text to model more complex implied dynamics, such as intuiting characters' mental states or predicting their plausible next actions.

In this paper, we construct a new annotation formalism to densely label commonsense short stories (Mostafazadeh et al., 2016) in terms of the mental states of the characters. The resulting dataset offers three unique properties. First, as highlighted in Figure 1, the dataset provides a fully-specified chain of *motivations* and *emotional reactions* for each story character as pre- and post-conditions of events. Second, the annotations include state changes for entities even when they are not mentioned directly in a sentence (e.g., in the fourth sentence in Figure 1, players would feel *afraid* as a result of the instructor throwing a chair), thereby capturing implied effects unstated in the story. Finally, the annotations encompass both formal labels from multiple theories of psychology (Maslow, 1943; Reiss, 2004; Plutchik, 1980) as well as open text descriptions of motivations and emotions, providing a comprehensive mapping between open text explanations and label categories (e.g., "to spend time with her son"

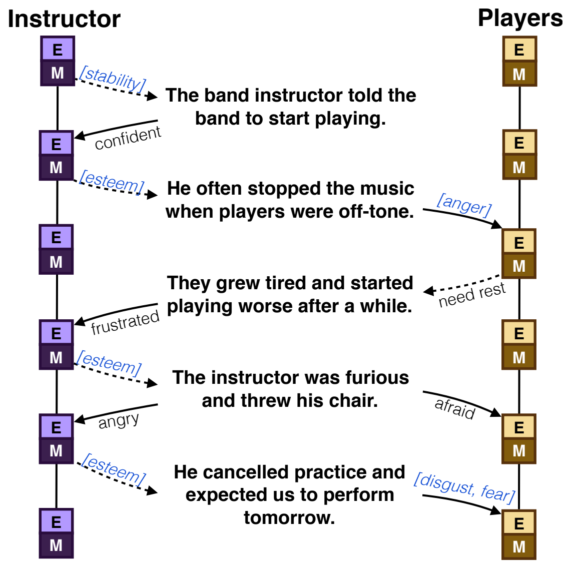

Instructor
Players
E
M
[stability]
The band instructor told the band to start playing.
confident
[esteem]
He often stopped the music when players were off-tone.
[anger]
They grew tired and started playing worse after a while.
need rest
frustrated
[esteem]
The instructor was furious and threw his chair.
afraid
angry
[esteem]
He cancelled practice and expected us to perform tomorrow.
[disgust, fear]

Figure 1: A story example with partial annotations for motivations (dashed) and emotional reactions (solid). Open text explanations are in black (e.g., "frustrated") and formal theory labels are in blue with brackets (e.g., "[esteem]").

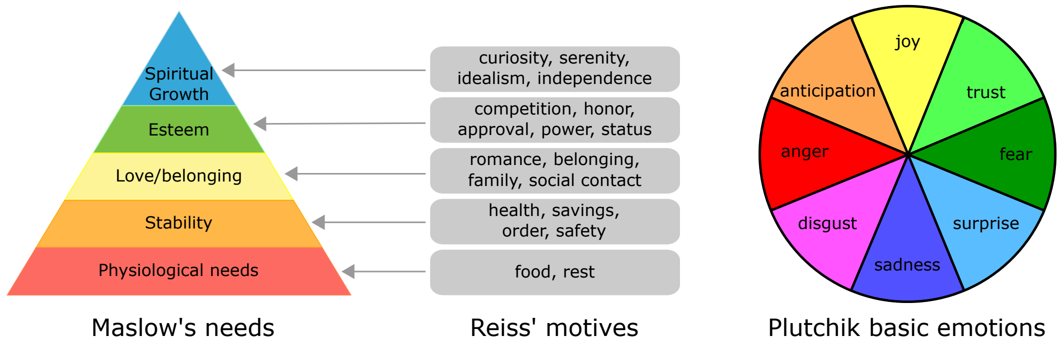


Figure 2: Theories of Motivation (Maslow and Reiss) and Emotional Reaction (Plutchik).

→ Maslow's category *love*). Our corpus[1] spans across 15k stories, amounting to 300k low-level annotations for around 150k character-line pairs.

Using our new corpus, we present baseline performance on two new tasks focusing on mental state tracking of story characters: *categorizing* motivations and emotional reactions using theory labels, as well as *describing* motivations and emotional reactions using open text. Empirical results demonstrate that existing neural network models including those with explicit or latent entity representations achieve promising results.

2 Mental State Representations

Understanding people's actions, motivations, and emotions has been a recurring research focus across several disciplines including philosophy and psychology (Schachter and Singer, 1962; Burke, 1969; Lazarus, 1991; Goldman, 2015). We draw from these prior works to derive a set of categorical labels for annotating the step-by-step causal dynamics between the mental states of story characters and the events they experience.

2.1 Motivation Theories

We use two popular theories of motivation: the "hierarchy of needs" of Maslow (1943) and the "basic motives" of Reiss (2004) to compile 5 coarse-grained and 19 fine-grained motivation categories, shown in Figure 2. Maslow's "hierarchy of needs" are comprised of five categories, ranging from *physiological needs* to *spiritual growth*, which we use as coarse-level categories. Reiss (2004) proposes 19 more fine-grained categories that provide a more informative range of motivations. For example, even though they both relate to the *physiological needs* Maslow category, the *food* and *rest* motives from Reiss (2004) are very different. While the Reiss theory allows for finer-grained annotations of motivation, the larger set of abstract concepts can be overwhelming for annotators. Motivated by Straker (2013), we design a hybrid approach, where Reiss labels are annotated as sub-categories of Maslow categories.

2.2 Emotion Theory

Among several theories of emotion, we work with the "wheel of emotions" of Plutchik (1980), as it has been a common choice in prior literature on emotion categorization (Mohammad and Turney, 2013; Zhou et al., 2016). We use the eight basic emotional dimensions as illustrated in Figure 2.

2.3 Mental State Explanations

In addition to the motivation and emotion categories derived from psychology theories, we also obtain open text descriptions of character mental states. These open text descriptions allow learning computational models that can *explain* the mental states of characters in natural language, which is likely to be more accessible and informative to end users than having theory categories alone. Collecting both theory categories and open text also allows us to learn the automatic mappings between the two, which generalizes the previous work of Mohammad and Turney (2013) on emotion category mappings.

3 Annotation Framework

In this study, we choose to annotate the simple commonsense stories introduced by Mostafazadeh et al. (2016). Despite their simplicity, these stories pose a significant challenge to natural language understanding models (Mostafazadeh et al., 2017).

[1]We make our dataset publicly available at `https://uwnlp.github.io/storycommonsense/`

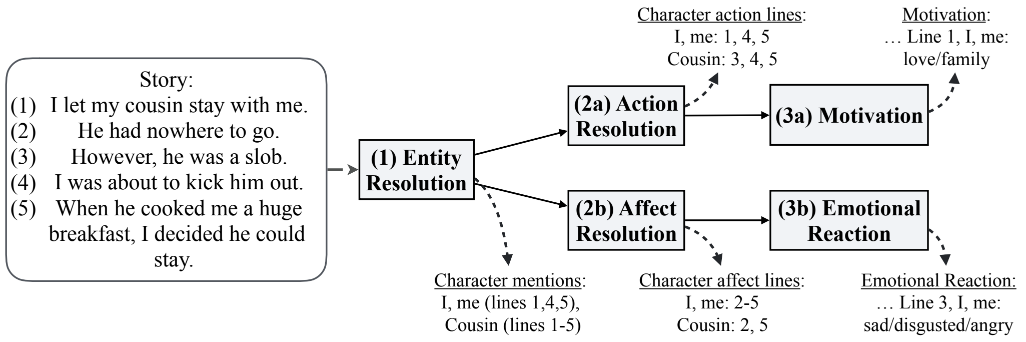


Figure 3: The annotation pipeline for the fine-grained annotations with an example story.

In addition, they depict multiple interactions between story characters, presenting rich opportunities to reason about character motivations and reactions. Furthermore, there are more than 98k such stories currently available covering a wide range of everyday scenarios.

Unique Challenges While there have been a variety of annotated resources developed on the related topics of sentiment analysis (Mohammad and Turney, 2013; Deng and Wiebe, 2015), entity tracking (Hoffart et al., 2011; Weston et al., 2015), and story understanding (Goyal et al., 2010; Ouyang and McKeown, 2015; Lukin et al., 2016), our study is the first to annotate the full chains of mental state effects for story characters. This poses several unique challenges as annotations require (1) interpreting discourse (2) understanding implicit causal effects, and (3) understanding formal psychology theory categories. In prior literature, annotations of this complexity have typically been performed by experts (Deng and Wiebe, 2015; Ouyang and McKeown, 2015). While reliable, these annotations are prohibitively expensive to scale up. Therefore, we introduce a new annotation framework that pipelines a set of smaller isolated tasks as illustrated in Figure 3. All annotations were collected using crowdsourced workers from Amazon Mechanical Turk.

3.1 Annotation Pipeline

We describe the components and workflow of the full annotation pipeline shown in Figure 3 below. The example story in the figure is used to illustrate the output of various steps in the pipeline (full annotations for this example are in the appendix).

(1) Entity Resolution The first task in the pipeline aims to discover (1) the set of characters E_i in each story i and (2) the set of sentences S_{ij} in which a specific character $j \in E_i$ is explicitly mentioned. For example, in the story in Figure 3, the characters identified by annotators are "I/me" and "My cousin", whom appear in sentences $\{1, 4, 5\}$ and $\{1, 2, 3, 4, 5\}$, respectively.

We use S_{ij} to control the workflow of later parts of the pipeline by pruning future tasks for sentences that are not tied to characters. Because S_{ij} is used to prune follow-up tasks, we take a high recall strategy to include all sentences that at least one annotator selected.

(2a) Action Resolution The next task identifies whether a character j appearing in a sentence k is taking any action to which a motivation can be attributed. We perform action resolution only for sentences $k \in S_{ij}$. In the running example, we would want to know that the cousin in line 2 is not doing anything intentional, allowing us to omit this line in the next pipeline stage (3a) where a character's motives are annotated. Description of state (e.g., "Alex is feeling blue") or passive event participation (e.g., "Alex trips") are not considered volitional acts for which the character may have an underlying motive. For each line and story character pair, we obtain 4 annotations. Because pairs can still be filtered out in the next stage of annotation, we select a generous threshold where only 2 annotators must vote that an intentional action took place for the sentence to be used as an input to the motivation annotation task (3a).

(2b) Affect Resolution This task aims to identify all of the lines where a story character j has an emotional reaction. Importantly, it is often possible to infer the emotional reaction of a character j even when the character does not explicitly appear in a sentence k. For instance, in Figure 3, we want to annotate the narrator's reaction to line 2 even though they are not mentioned because their emotional response is inferrable. We obtain 4 an-

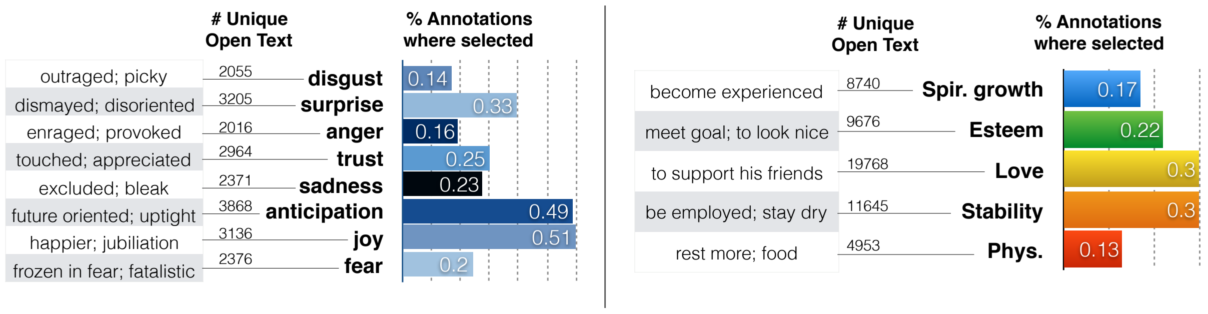


Figure 4: Examples of open-text explanations that annotators provided corresponding with the categories they selected. The bars on the right of the categories represent the percentage of lines where annotators selected that category (out of those character-line pairs with positive motivation/emotional reaction).

notations per character per line. The lines with at least 2 annotators voting are used as input for the next task: (3b) emotional reaction.

(3a) Motivation We use the output from the action resolution stage (2a) to ask workers to annotate character motives in lines where they intentionally initiate an event. We provide 3 annotators a line from a story, the preceding lines, and a specific character. They are asked to produce a free response sentence describing what causes the character's behavior in that line and to select the most related Maslow categories and Reiss subcategories. In Figure 3, an annotator described the motivation of the narrator in line 1 as wanting "to have company" and then selected the *love* (Maslow) and *family* (Reiss) as categorical labels. Because many annotators are not familiar with motivational theories, we require them to complete a tutorial the first time they attempt the task.

(3b) Emotional Reaction Simultaneously, we use the output from the affect resolution stage (2b) to ask workers what the emotional response of a character is immediately following a line in which they are affected. As with the motives, we give 3 annotators a line from a story, its previous context, and a specific character. We ask them to describe in open text how the character will feel following the event in the sentence (up to three emotions). As a follow-up, we ask workers to compare their free responses against Plutchik categories by using 3-point likert ratings. In Figure 3, we include a response for the emotional reaction of the narrator in line 1. Even though the narrator was not mentioned directly in that line, an annotator recorded that they will react to their cousin being a slob by feeling "annoyed" and selected the Plutchik categories for *sadness*, *disgust* and *anger*.

		Fine-grained	
	train	dev	test
# annotated stories	10000	2500	2500
# characters / story	2.03	2.02	1.82
# char-lines w/ motiv	40154	8762	6831
# char-lines w/ emot	76613	14532	13785

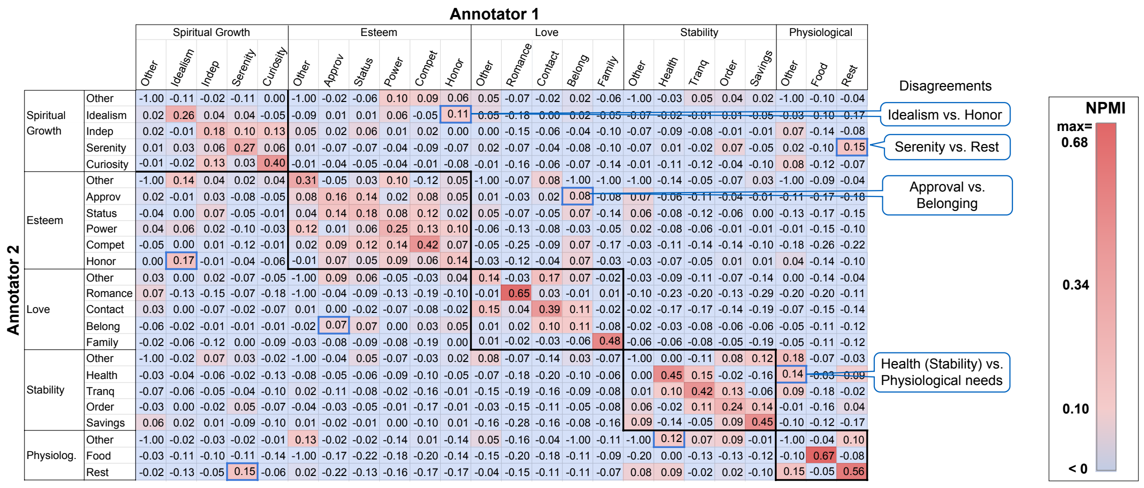
Table 1: Annotated data statistics for each dataset

3.2 Dataset Statistics and Insights

Cost The tasks corresponding to the theory category assignments are the hardest and most expensive in the pipeline (∼\$4 per story). Therefore, we obtain theory category labels only for a third of our annotated stories, which we assign to the development and test sets. The training data is annotated with a shortened pipeline with only open text descriptions of motivations and emotional reactions from two workers (∼\$1 per story).

Scale Our dataset to date includes a total of 300k low-level annotations for motivation and emotion across 15,000 stories (randomly selected from the ROC story training set). It covers over 150,000 character-line pairs, in which 56k character-line pairs have an annotated motivation and 105k have an annotated change in emotion (i.e. a label other than `none`). Table 1 shows the break down across training, development, and test splits. Figure 4 shows the frequency of different labels being selected for motivational and emotional categories in cases with positive change.

Agreements For quality control, we removed workers who consistently produced low-quality work, as discussed in the Appendix. In the categorization sets (Maslow, Reiss and Plutchik), we compare the performance of annotators by treating each individual category as a binary label (1


Annotator 1
Annotator 2
Disagreements
Idealism vs. Honor
Serenity vs. Rest
Approval vs. Belonging
Health (Stability) vs. Physiological needs
NPMI
max= 0.68
0.34
0.10
< 0

		Spiritual Growth					Esteem						Love					Stability					Physiological		
		Other	Idealism	Indep	Serenity	Curiosity	Other	Approv	Status	Power	Compet	Honor	Other	Romance	Contact	Belong	Family	Other	Health	Tranq	Order	Savings	Other	Food	Rest
Spiritual Growth	Other	-1.00	-0.11	-0.02	-0.11	0.00	-1.00	-0.02	-0.06	0.10	0.09	0.06	0.05	-0.07	-0.02	0.02	-0.06	-1.00	-0.03	0.05	0.04	0.02	-1.00	-0.10	-0.04
	Idealism	0.02	0.26	0.04	0.04	-0.05	-0.09	0.01	0.01	0.06	-0.05	0.11	-0.05	-0.18	0.00	0.02	-0.05	-0.07	-0.02	-0.01	0.01	-0.05	-0.03	-0.10	-0.17
	Indep	0.02	-0.01	0.18	0.10	0.13	0.05	0.02	0.06	0.01	0.02	0.00	0.00	-0.15	-0.06	-0.04	-0.10	-0.07	-0.09	-0.08	-0.01	-0.01	0.07	-0.14	-0.08
	Serenity	0.01	0.03	0.06	0.27	0.06	0.01	-0.07	-0.07	-0.04	-0.09	-0.07	0.02	-0.07	-0.04	-0.08	-0.10	-0.07	0.01	-0.02	0.07	-0.05	0.02	-0.10	0.15
	Curiosity	-0.01	-0.02	0.13	0.03	0.40	-0.01	-0.04	-0.05	-0.04	-0.01	-0.08	-0.01	-0.16	-0.06	-0.07	-0.14	-0.01	-0.11	-0.12	-0.04	-0.10	0.08	-0.12	-0.07
Esteem	Other	-1.00	0.14	0.04	0.02	0.04	0.31	-0.05	0.03	0.10	-0.12	0.05	-1.00	-0.07	0.08	-1.00	-1.00	-1.00	-0.14	-0.05	-0.07	0.03	-1.00	-0.09	-0.04
	Approv	0.02	-0.01	0.03	-0.08	-0.05	0.08	0.16	0.14	0.02	0.08	0.05	0.01	-0.03	0.02	0.08	-0.08	0.07	-0.06	-0.11	-0.04	-0.01	-0.11	-0.17	-0.18
	Status	-0.04	0.00	0.07	-0.05	-0.01	0.04	0.14	0.18	0.08	0.12	0.02	0.05	-0.07	-0.05	0.07	-0.14	0.06	-0.08	-0.12	-0.06	0.00	-0.13	-0.17	-0.15
	Power	0.04	0.06	0.02	-0.10	-0.03	0.12	0.01	0.06	0.25	0.13	0.10	-0.06	-0.13	-0.08	-0.03	-0.05	0.02	-0.08	-0.06	-0.01	-0.01	-0.01	-0.15	-0.10
	Compet	-0.05	0.00	0.01	-0.12	-0.01	0.02	0.09	0.12	0.14	0.42	0.07	-0.05	-0.25	-0.09	0.07	-0.17	-0.03	-0.11	-0.14	-0.14	-0.10	-0.18	-0.26	-0.22
	Honor	0.00	0.17	-0.01	-0.04	-0.06	-0.01	0.07	0.05	0.09	0.06	0.14	-0.03	-0.12	-0.04	0.07	-0.03	-0.03	-0.07	-0.05	0.01	0.01	0.04	-0.14	-0.10
Love	Other	0.03	0.00	0.02	-0.07	-0.05	-1.00	0.09	0.06	-0.05	-0.03	0.04	0.14	-0.03	0.17	0.07	-0.02	-0.03	-0.09	-0.11	-0.07	-0.14	0.00	-0.14	-0.04
	Romance	0.07	-0.13	-0.15	-0.07	-0.18	-1.00	-0.04	-0.09	-0.13	-0.19	-0.10	-0.01	0.65	0.03	-0.01	0.01	-0.10	-0.23	-0.20	-0.13	-0.29	-0.20	-0.20	-0.11
	Contact	0.03	0.00	-0.07	-0.02	-0.07	0.01	0.00	-0.02	-0.07	-0.08	-0.02	0.15	0.04	0.39	0.11	-0.02	-0.02	-0.17	-0.17	-0.14	-0.19	-0.07	-0.15	-0.14
	Belong	-0.06	-0.02	-0.01	-0.01	-0.01	-0.02	0.07	0.07	0.00	0.03	0.05	0.01	0.02	0.10	0.11	-0.08	-0.02	-0.03	-0.08	-0.06	-0.06	-0.05	-0.11	-0.12
	Family	-0.02	-0.06	-0.12	0.00	-0.09	-0.03	-0.08	-0.09	-0.08	-0.19	0.00	0.01	-0.02	-0.03	-0.06	0.48	-0.06	-0.06	-0.08	-0.05	-0.19	-0.05	-0.11	-0.12
Stability	Other	-1.00	-0.02	0.07	0.03	-0.02	-1.00	-0.04	0.05	-0.07	-0.03	0.02	0.08	-0.07	-0.14	0.03	-0.07	-1.00	0.00	-0.11	0.08	0.12	0.18	-0.07	-0.03
	Health	-0.03	-0.04	-0.06	-0.02	-0.13	-0.08	-0.05	-0.06	-0.09	-0.10	-0.05	-0.07	-0.18	-0.20	-0.10	-0.06	0.00	0.45	0.15	-0.02	-0.16	0.14	-0.03	0.09
	Tranq	-0.07	-0.06	-0.05	-0.04	-0.10	0.02	-0.11	-0.08	-0.02	-0.16	-0.01	-0.15	-0.19	-0.16	-0.09	-0.08	0.01	0.10	0.42	0.13	-0.06	0.09	-0.18	-0.02
	Order	-0.03	0.00	-0.02	0.05	-0.07	-0.04	-0.03	-0.05	-0.01	-0.17	-0.01	-0.03	-0.15	-0.11	-0.05	-0.08	0.06	-0.02	0.11	0.24	0.14	-0.01	-0.16	0.04
	Savings	0.06	0.02	0.01	-0.09	-0.10	0.01	-0.02	-0.01	0.00	-0.10	0.01	-0.16	-0.28	-0.16	-0.08	-0.16	0.09	-0.14	-0.05	0.09	0.45	-0.10	-0.12	-0.17
Physiolog.	Other	-1.00	-0.02	-0.03	-0.02	-0.01	0.13	-0.02	-0.02	-0.14	0.01	-0.14	0.05	-0.16	-0.04	-1.00	-0.11	-1.00	0.12	0.07	0.09	-0.01	-1.00	-0.04	0.10
	Food	-0.03	-0.11	-0.10	-0.11	-0.14	-1.00	-0.17	-0.22	-0.18	-0.20	-0.14	-0.15	-0.20	-0.18	-0.11	-0.09	-0.20	0.00	-0.13	-0.13	-0.12	-0.10	0.67	-0.08
	Rest	-0.02	-0.13	-0.05	0.15	-0.06	0.02	-0.22	-0.13	-0.16	-0.17	-0.17	-0.04	-0.15	-0.11	-0.11	-0.07	0.08	0.09	-0.02	0.02	-0.10	0.15	-0.05	0.56

Figure 5: NPMI confusion matrix on motivational categories for all annotator pairs with color scaling for legibility. The highest values are generally along diagonal or within Maslow categories (outlined in black). We highlight a few common points of disagreement between thematically similar categories.

Label Type		**PPA**	**KA**	**% Agree w/ Maj. Lbl**
Maslow	Dev	.77	.30	0.88
	Test	.77	.31	0.89
Reiss	Dev	.91	.24	0.95
	Test	.91	.24	0.95
Plutchik	Dev	.71	.32	0.84
	Test	.70	.29	0.83

Table 2: Agreement Statistics (PPA = Pairwise percent agreement of worker responses per binary category, KA= Krippendorff's Alpha)

if they included the category in their set of responses) and averaging the agreement per category. For Plutchik scores, we count 'moderately associated' ratings as agreeing with 'highly' associated' ratings. The percent agreement and Krippendorff's alpha are shown in Table 2. We also compute the percent agreement between the individual annotations and the majority labels.[2]

These scores are difficult to interpret by themselves, however, as annotator agreement in our categorization system has a number of properties that are not accounted for by these metrics (disagreement preferences – joy and trust are closer than joy and anger – that are difficult to quantify in a principled way, hierarchical categories mapping Reiss subcategories from Maslow categories, skewed category distributions that inflate PPA and deflate KA scores, and annotators that could select multiple labels for the same examples).

To provide a clearer understanding of agreement within this dataset, we create aggregated confusion matrices for annotator pairs. First, we sum the counts of combinations of answers between all paired annotations (excluding `none` labels). If an annotator selected multiple categories, we split the count uniformly among the selected categories. We compute NPMI over the total confusion matrix. In Figure 5, we show the NPMI confusion matrix for motivational categories.

In the motivation annotations, we find the highest scores on the diagonal (i.e., Reiss agreement), with most confusions occurring between Reiss motives in the same Maslow category (outlined black in Figure 5). Other disagreements generally involve Reiss subcategories that are thematically similar, such as *serenity* (mental relaxation) and *rest* (physical relaxation). We provide this analysis for Plutchik categories in the appendix, finding high scores along the diagonal with disagreements typically occurring between categories in a "positive emotion" cluster (*joy*, *trust*) or a "negative emotion" cluster (*anger*, *disgust*,*sadness*).

4 Tasks

The multiple modes covered by the annotations in this new dataset allow for multiple new tasks to be explored. We outline three task types below, covering a total of eight tasks on which to evaluate.

[2]Majority label for the motivation categories is what was agreed upon by at least two annotators per category. For emotion categories, we averaged the point-wise ratings and counted a category if the average rating was ≥ 2.

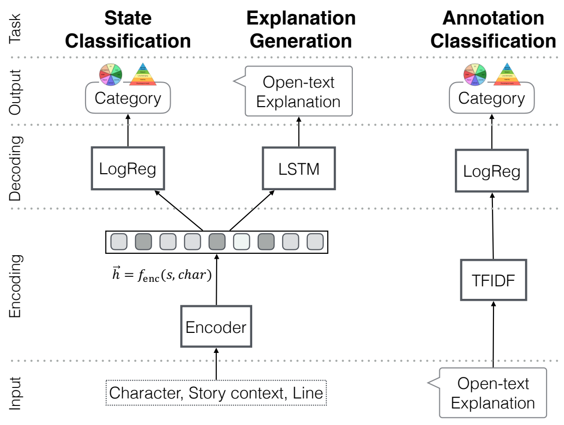


Figure 6: General model architectures for three new task types

Differences between task type inputs and outputs are summarized in Figure 6.

State Classification The three primary tasks involve categorizing the psychological states of story characters for each of the label sets (Maslow, Reiss, Plutchik) collected for the dev and test splits of our dataset. In each classification task, a model is given a line of the story (along with optional preceding context lines) and a character and predicts the motivation (or emotional reaction). A binary label is predicted for each of the Maslow needs, Reiss motives or Plutchik categories.

Annotation Classification Because the dev and test sets contain paired classification labels and free text explanations, we propose three tasks where a model must predict the correct Maslow/Reiss/Plutchik label given an emotional reaction or motivation explanation.

Explanation Generation Finally, we can use the free text explanations to train models to describe the psychological state of a character in free text (examples in Figure 4). These explanations allow for two conditional generation tasks where the model must generate the words describing the emotional reaction or motivation of the character.

5 Baseline Models

The general model architectures for the three tasks are shown in Figure 6. We describe each model component below. The state classification and explanation generation models could be trained separately or in a multi-task set-up.

In the state classification and explanation generation tasks, a model is given a line from a story $\mathbf{x}^s$ containing N words $\{w_0^s, w_1^s, \ldots, w_N^s\}$ from vocabulary V, a character in that story $e_j \in E$ where E is the set of characters in the story, and (optionally) the preceding sentences in the story $\mathbf{C} = \{\mathbf{x}^0 \ldots, \mathbf{x}^{s-1}\}$ containing words from vocabulary V. A representation for a character's psychological state is encoded as:

$$\mathbf{h}^e = \text{Encoder}(\mathbf{x}^s, \mathbf{C}[e_j]) \tag{1}$$

where $\mathbf{C}[e_j]$ corresponds to the concatenated subset of sentences in $\mathbf{C}$ where e_j appears.

5.1 Encoders

While the end classifier or decoder is different for each task, we use the same set of encoders based on word embeddings, common neural network architectures, or memory networks to formulate a representation of the sentence and character, $\mathbf{h}^e$. Unless specified, $\mathbf{h}^e$ is computed by encoding separate vector representations for the sentence ($\mathbf{x}^s \rightarrow \mathbf{h}^s$) and character-specific context ($\mathbf{C}[e_j] \rightarrow \mathbf{h}^c$) and concatenating these encodings ($\mathbf{h}^e = [\mathbf{h}^c; \mathbf{h}^s]$). We describe the encoders below:

TF-IDF We learn a TD-IDF model on the full training corpus of Mostafazadeh et al. (2016) (excluding the stories in our dev/test sets). To encode the sentence, we extract TF-IDF features for its words, yielding $v^s \in \mathcal{R}^V$. A projection and non-linearity is applied to these features to yield $\mathbf{h}^s$:

$$\mathbf{h}^s = \phi(W_s v^s + b_s) \tag{2}$$

where $W_s \in \mathcal{R}^{d \times H}$. The character vector $\mathbf{h}^c$ is encoded in the same way on sentences in the context pertaining to the character.

GloVe We extract pretrained Glove vectors (Pennington et al., 2014) for each word in V. The word embeddings are max-pooled, yielding embedding $v^s \in \mathcal{R}^H$, where H is the dimensionality of the Glove vectors. Using this max-pooled representation, $\mathbf{h}^s$ and $\mathbf{h}^c$ are extracted in the same manner as for TF-IDF features (Equation 2).

CNN We implement a CNN text categorization model using the same configuration as Kim (2014) to encode the sentence words. A sentence is represented as a matrix, $v^s \in \mathcal{R}^{M \times d}$ where each row is a word embedding x_n^s for a word $w_n^s \in \mathbf{x}^s$.

$$v^s = [x_0^s, x_1^s, \ldots, x_N^s] \tag{3}$$

$$\mathbf{h}^s = \text{CNN}(v^s) \tag{4}$$

where CNN represents the categorization model from (Kim, 2014). The character vector $\mathbf{h}^c$ is encoded in the same way with a separate CNN. Implementation details are provided in the appendix.

LSTM A two-layer bi-LSTM encodes the sentence words and concatenates the final time step hidden states from both directions to yield $\mathbf{h}^s$. The character vector $\mathbf{h}^c$ is encoded the same way.

REN We use the "tied" recurrent entity network from Henaff et al. (2017). A memory cell m is initialized for each of the J characters in the story, $E = \{e_0, \ldots, e_J\}$. The REN reads documents one sentence at a time and updates m_j for $e_j \in E$ after reading each sentence. Unlike the previous encoders, all sentences of the context $\mathbf{C}$ are given to the REN along with the sentence $\mathbf{x}^s$. The model learns to distribute encoded information to the correct memory cells. The representation passed to the downstream model is:

$$\mathbf{h}^e = \{m_j\}^s \quad (5)$$

where $\{m_j\}^s$ is the memory vector in the cell corresponding to e_j after reading $\mathbf{x}^s$. Implementation details are provided in the appendix.

NPN We also include the neural process network from Bosselut et al. (2018) with "tied" entities, but "untied" actions that are not grounded to particular concepts. The memory is initialized and accessed similarly as the REN. Exact implementation details are provided in the appendix.

5.2 State Classifier

Once the sentence-character encoding $\mathbf{h}^e$ is extracted, the state classifier predicts a binary label $\hat{y}_z$ for every category $z \in \mathcal{Z}$ where $\mathcal{Z}$ is the set of category labels for a particular psychological theory (e.g., disgust, surprise, etc. in the Plutchik wheel). We use logistic regression as a classifier:

$$\hat{y}_z = \sigma(W_z \mathbf{h}^e + b_z) \quad (6)$$

where W_z and b_z are a label-specific set of weights and biases for classifying each label $z \in \mathcal{Z}$.

5.3 Explanation Generator

The explanation generator is a single-layer LSTM (Hochreiter and Schmidhuber, 1997) that receives the encoded sentence-character representation $\mathbf{h}^e$ and predicts each word y_t in the explanation using the same method from Sutskever et al. (2014). Implementation details are provided in the appendix.

5.4 Annotation Classifier

For annotation classification tasks, words from open-text explanations are encoded with TF-IDF features. The same classifier architecture from Section 5.2 is used to predict the labels.

6 Experimental Setup

6.1 Training

State Classification The dev set D is split into two portions of 80% (D_1) and 20% (D_2). D_1 is used to train the classifier and encoder. D_2 is used to tune hyperparameters. The model is trained to minimize the weighted binary cross entropy of predicting a class label y_z for each class z:

$$\mathcal{L} = \sum_{z=1}^{Z} \gamma_z y_z \log \hat{y}_z + (1-\gamma_z)(1-y_z)\log(1-\hat{y}_z) \quad (7)$$

where Z is the number of labels in each of the three classifications tasks and γ_z is defined as:

$$\gamma_z = 1 - e^{-\sqrt{P(y_z)}} \quad (8)$$

where $P(y_z)$ is the marginal class probability of a positive label for z in the training set.

Annotation Classification The dev set is split in the same manner as for state classification. The TF-IDF features are trained on the set of training annotations D_t coupled with those from D_1. The model must minimize the same loss as in Equation 7. Details are provided in the appendix.

Explanation Generation We use the training set of open annotations to train a model to predict explanations. The decoder is trained to minimize the negative loglikelihood of predicting each word in the explanation of a character's state:

$$L_{gen} = -\sum_{t=1}^{T} \log P(y_t | y_0, ..., y_{t-1}, \mathbf{h}^e) \quad (9)$$

where $\mathbf{h}^e$ is the sentence-character representation produced by an encoder from Section 5.1.

6.2 Metrics

Classification For the state and annotation classification task, we report the micro-averaged precision (P), recall (R), and F1 score of the Plutchik, Maslow, and Reiss prediction tasks. We report the results of selecting a label at random in the top two rows of Table 3. Note that random is low because the distribution of positive instances for each

Model	Maslow			Reiss			Plutchik		
	P	R	F1	P	R	F1	P	R	F1
Random	7.45	49.99	12.96	1.76	50.02	3.40	10.35	50.00	17.15
Random (Weighted)	8.10	8.89	8.48	2.25	2.40	2.32	12.28	11.79	12.03
TF-IDF	30.10	21.21	24.88	18.40	20.67	19.46	20.05	24.11	21.90
+ Entity Context	29.79	34.56	32.00	20.55	24.81	22.48	22.71	25.24	23.91
GloVe	25.15	29.70	27.24	16.65	18.83	17.67	15.19	30.56	20.29
+ Entity Context	27.02	37.00	31.23	16.99	26.08	20.58	19.47	**46.65**	27.48
LSTM	24.64	35.30	29.02	19.91	19.76	19.84	20.27	30.37	24.31
+ Entity Context	**31.29**	33.85	32.52	18.35	27.61	22.05	23.98	31.41	27.20
+ Explanation Training	30.34	40.12	34.55	**21.38**	28.70	**24.51**	25.31	33.44	28.81
CNN (Kim, 2014)	26.21	31.09	28.44	20.30	23.24	21.67	21.15	23.36	22.20
+ Entity Context	27.47	41.01	32.09	18.89	31.22	23.54	24.32	30.76	27.16
+ Explanation Training	29.30	44.18	**35.23**	17.87	**37.52**	24.21	24.47	38.87	30.04
REN (Henaff et al., 2017)	26.24	42.14	32.34	16.79	22.20	19.12	**26.22**	33.26	29.32
+ Explanation Training	26.85	**44.78**	33.57	16.73	26.55	20.53	25.30	37.30	30.15
NPN (Bosselut et al., 2018)	24.27	44.16	31.33	13.13	26.44	17.55	21.98	37.31	27.66
+ Explanation Training	26.60	39.17	31.69	15.75	20.34	17.75	24.33	40.10	**30.29**

Table 3: State Classification Results

category is very uneven: macro-averaged positive class probabilities of 8.2, 1.7, and 9.9% per category for Maslow, Reiss, and Plutchik respectively.

Generation Because explanations tend to be short sequences (Figure 4) with high levels of synonymy, traditional metrics such as BLEU are inadequate for evaluating generation quality. We use the vector average and vector extrema metrics from Liu et al. (2016) computed using the Glove vectors of generated and reference words. We report results in Table 5 on the dev set and compare to a baseline that randomly samples an example from the dev set as a generated sequence.

6.3 Ablations

Story Context vs. No Context Our dataset is motivated by the importance of interpreting story context to categorize emotional reactions and motivations of characters. To test this importance, we ablate $\mathbf{h}^c$, the representation of the context sentences pertaining to the character, as an input to the state classifier for each encoder (except the REN and NPN). In Table 3, this ablation is the first row for each encoder presented.

Explanation Pretraining Because the state classification and explanation generation tasks use the same models to encode the story, we explore initializing a classification encoder with parameters trained on the generation task. For the CNN, LSTM, and REN encoders, we pretrain a generator to produce emotion or motivation explanations. We use the parameters from the emotion or motivation explanation generators to initialize the Plutchik or Maslow/Reiss classifiers respectively.

7 Experimental Results

State Classification We show results on the test set for categorizing Maslow, Reiss, and Plutchik states in Table 3. Despite the difficulty of the task, all models outperform the random baseline. Interestingly, the performance boost from adding entity-specific contextual information (i.e., not ablating $\mathbf{h}^c$) indicates that the models learn to condition on a character's previous experience to classify its mental state at the current time step. This effect can be seen in a story about a man whose flight is cancelled. The model without context predicts the same emotional reactions for the man, his wife and the pilot, but with context correctly predicts that the pilot will not have a reaction while predicting that the man and his wife will feel sad.

For the CNN, LSTM, REN, and NPN models, we also report results from pretraining encoder parameters using the free response annotations from the training set. This pretraining offers a clear performance boost for all models on all three prediction tasks, showing that the parameters of the encoder can be pretrained on auxiliary tasks providing emotional and motivational state signal.

The best performing models in each task are most effective at predicting Maslow *physiological* needs, Reiss *food* motives, and Plutchik reactions of *joy*. The relative ease of predicting motivations

	Maslow	Reiss	Plutchik
TFIDF	64.81	48.60	53.44

Table 4: F1 scores of predicting correct category labels from free response annotations

Model	Motivation		Emotion	
	Avg	VE	Avg	VE
Random	56.02	45.75	40.23	39.98
LSTM	58.48	51.07	52.47	52.30
CNN	57.83	50.75	52.49	52.31
REN	**58.83**	**51.79**	53.95	53.79
NPN	57.77	51.77	**54.02**	**53.85**

Table 5: Vector average and extrema scores for generation of annotation explanations

related to food (and physiological needs generally) may be because they involve a more limited and concrete set of actions such as eating or cooking.

Annotation Classification Table 4 shows that a simple model can learn to map open text responses to categorical labels. This further supports our hypothesis that pretraining a classification model on the free-response annotations could be helpful in boosting performance on the category prediction.

Explanation Generation Finally, we provide results for the task of generating explanations of motivations and emotions in Table 5. Because the explanations are closely tied to emotional and motivation states, the randomly selected explanation can often be close in embedding space to the reference explanations, making the random baseline fairly competitive. However, all models outperform the strong baseline on both metrics, indicating that the generated short explanations are closer semantically to the reference annotation.

8 Related work

Mental State Annotations Incorporating emotion theories into NLP tasks has been explored in previous projects. Ghosh et al. (2017) modulate language model distributions by increasing the probability of words that express certain affective LIWC (Tausczik and Pennebaker, 2016) categories. More generally, various projects tackle the problem of generating text from a set of attributes like sentiment or generic-ness (Ficler and Goldberg, 2017; Dong et al., 2017). Similarly, there is also a body of research in reasoning about commonsense stories and discourse (Li and Jurafsky, 2017; Mostafazadeh et al., 2016) or detecting emotional stimuli in stories (Gui et al., 2017). Previous work in plot units (Lehnert, 1981) developed formalisms for affect and mental state in story narratives that included motivations and reactions. In our work, we collect mental state annotations for stories to used as a new resource in this space.

Modeling Entity State Recently, novel works in language modeling (Ji et al., 2017; Yang et al., 2016), question answering (Henaff et al., 2017), and text generation (Kiddon et al., 2016; Bosselut et al., 2018) have shown that modeling entity state explicitly can boost performance while providing a preliminary interface for interpreting a model's prediction. Entity modeling in these works, however, was limited to tracking entity reference (Kiddon et al., 2016; Yang et al., 2016; Ji et al., 2017), recognizing entity state similarity (Henaff et al., 2017) or predicting simple attributes from entity states (Bosselut et al., 2018). Our work provides a new dataset for tracking emotional reactions and motivations of characters in stories.

9 Conclusion

We present a large scale dataset as a resource for training and evaluating mental state tracking of characters in short commonsense stories. This dataset contains over 300k low-level annotations for character *motivations* and *emotional reactions*. We provide benchmark results on this new resource. Importantly, we show that modeling character-specific context and pretraining on free-response data can boost labeling performance. While our work only use information present in our dataset, we view our dataset as a future testbed for evaluating models trained on any number of resources for learning common sense about emotional reactions and motivations.

Acknowledgments

We thank the reviewers for their insightful comments. We also thank Bowen Wang, xlab members, Martha Palmer, Tim O'Gorman, Susan W. Brown, and Ghazaleh Kazeminejad for helpful discussions on inter-annotator agreement and the annotation pipeline. This work was supported in part by NSF GRFP DGE-1256082, NSF IIS-1714566, IIS-1524371, Samsung AI, and DARPA CwC (W911NF-15-1-0543).

References

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. In *Proceedings of the 6th International Conference on Learning Representations*.

Kenneth Burke. 1969. *A Grammar of Motives*. Univ of California Press.

Lingjia Deng and Janyce Wiebe. 2015. Mpqa 3.0: An entity/event-level sentiment corpus. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1323–1328.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *EACL*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *EMNLP Workshop on Stylistic Variation*.

Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2017. Affect-LM: A neural language model for customizable affective text generation. In *ACL*.

Alvin I Goldman. 2015. *Theory of Human Action*. Princeton University Press.

Amit Goyal, Ellen Riloff, and Hal Daumé. 2010. Automatically producing plot unit representations for narrative text. In *EMNLP*.

Lin Gui, Jiannan Hu, Yulan He, Ruifeng Xu, Lu Qin, and Jiachen Du. 2017. A question answering approach for emotion cause extraction. In *EMNLP*. Association for Computational Linguistics, Copenhagen, Denmark, pages 1594–1603.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2017. Tracking the world state with recurrent entity networks. In *ICLR*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *EMNLP*.

Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A Smith. 2017. Dynamic entity representations in neural language models. In *EMNLP*. Association for Computational Linguistics, Copenhagen, Denmark, pages 1831–1840.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *EMNLP*. pages 329–339.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Richard S Lazarus. 1991. Progress on a cognitive-motivational-relational theory of emotion. *American Psychologist* 46(8):819.

Wendy G. Lehnert. 1981. Plot units and narrative summarization. *Cognitive Science* 5:293–331.

Jiwei Li and Dan Jurafsky. 2017. Neural net models for Open-Domain discourse coherence. In *EMNLP*.

Chia-Wei Liu, Ryan Joseph Lowe, Iulian 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. In *EMNLP*.

Stephanie M. Lukin, Kevin Bowden, Casey Barackman, and Marilyn A. Walker. 2016. Personabank: A corpus of personal narratives and their story intention graphs. *CoRR* abs/1708.09082.

Abraham H Maslow. 1943. A theory of human motivation. *Psychol. Rev.* 50(4):370.

Saif Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence* 29:436–465.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *NAACL*.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. Lsdsem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*. pages 46–51.

Jessica Ouyang and Kathleen McKeown. 2015. Modeling reportable events as turning points in narrative. In *EMNLP*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP*. pages 1532–1543.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. *Theories of emotion* 1(3-31):4.

Steven Reiss. 2004. Multifaceted nature of intrinsic motivation: The theory of 16 basic desires. *Rev. Gen. Psychol.* 8(3):179.

Stanley Schachter and Jerome Singer. 1962. Cognitive, social, and physiological determinants of emotional state. *Psychological review* 69(5):379.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

David Straker. 2013. Reiss' 16 human needs. `http://changingminds.org/explanations/needs/reiss_16_needs.htm`. Accessed: 2018-02-21.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*. pages 3104–3112.

Yla R Tausczik and James W Pennebaker. 2016. The psychological meaning of words: LIWC and computerized text analysis methods. *J. Lang. Soc. Psychol.* .

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698* .

Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. 2016. Reference-aware language models. *CoRR* abs/1611.01628. http://arxiv.org/abs/1611.01628.

Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. 2016. Situation recognition: Visual semantic role labeling for image understanding. In *Conference on Computer Vision and Pattern Recognition*.

Deyu Zhou, Xuan Zhang, Yin Zhou, Quanming Zhao, and Xin Geng. 2016. Emotion distribution learning from texts. In *EMNLP*.

A AMT Annotation Notes

We ran AMT tasks in batches. We performed quality control tests after every 1-2 batches and banned workers who were consistently performing poorly (and removed their responses from our dataset). For quality control, we used a combination of automatic methods (SGD over interannotator agreement using the quality control objectives described in the supplementary material of Yatskar et al., 2016) and manual methods (e.g. searching for workers who always selected the same answers, who had strange answering patterns, who copied-and-pasted free responses) to identify workers with poor-quality work. Work by such annotators was reviewed, and these workers were banned if they were repeatedly not following instructions on a number of HIT's.

B Emotion Agreement Matrix

We include a NPMI confusion matrix for aggregated Plutchik paired responses in Figure 7. Black boxes signify the same emotion but at different intensities (high vs. moderate). In general higher co-occurring responses are along the diagonal. However, we note that there are two main clusters coinciding with strongly positive emotions (joy and trust) and strongly negative emotions (anger, disgust, and sadness) where disagreements are more likely to occur. To a lesser extent, there also is a slight co-occurrence between fear and the strongly negative emotions.

C Model Implementation Details

All classification models are trained with the Adam Optimizer (Kingma and Ba, 2015) with a learning rate 0.001 and gradient clipping if the norm of the gradients exceeds 1. We regularize with dropout layers whose probabilities are specific to each model. All models are trained with word embeddings of dimensionality 100 that are initialized with pretrained Glove vectors (Pennington et al., 2014). For classification labels, we use the majority label among annotators for a particular character-line pair.

C.1 LSTM Classifier

We train a 2-layer bidirectional LSTM encoder. The hidden states of the LSTM have dimensionality 100. We add dropout layers with p=0.5 in between the word embedding layer and the LSTM and between LSTM layers (Srivastava et al., 2014). We include a dropout layer with p=0.5 before the logistic regression classifier.

C.2 CNN Classifier

We follow the approach of Kim (2014) and train a CNN classifier with kernels of size 3, 4, and 5. We use 100 kernels of each size. We add a dropout layer with p=0.5 between the word embedding layer and the convolutional kernel layers. We include a dropout layer with p=0.5 before the logistic regression classifier.

C.3 REN Classifier

We use the same implementation as Henaff et al. (2017) except that we remove the output module designed to encode questions and instead select the memory cell tied to the entity of interest for every training example. All equations from the input encoder and dynamic memory are identical to those of (Henaff et al., 2017). The input encoder computes a positionally weighted average of all the words in a sentence:

$$s_t = \sum_i f_i \odot e_i \tag{10}$$

where e_i is a word embedding at index i in a sentence, f_i is a positional weight for that index in the sentence, and s_t is a sentence representation. The dynamic memory is updated in the following way:

$$g_j = \sigma(s_t^T h_j + s_t^T w_j) \tag{11}$$

$$\tilde{h}_j = \phi(U h_j + V w_j + W s_t) \tag{12}$$

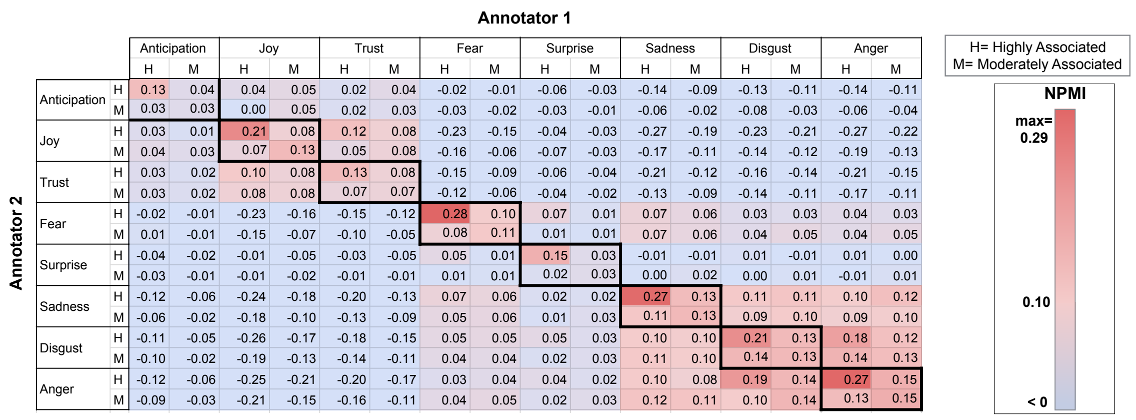
$$h_j \leftarrow h_j + g_j \odot \tilde{h}_j \tag{13}$$

$$h_j \leftarrow \frac{h_j}{||h_j||} \tag{14}$$

where h_j is the value stored in memory cell j, w_j is a key corresponding to memory cell j, U, V, W are projection matrices, and ϕ is a PReLU non-linearity. We initialize entity memory keys and entity memory values with the sum of the Glove vectors for all the words in the character name. Entity key values w_j are locked during training. We use dropout with p=0.3 between the encoder and dynamic memory.

C.4 NPN Classifier

We use the same implementation as in Bosselut et al. (2018) with a few modifications to the underlying architecture. First, we use the same encoder


Annotator 1
Annotator 2
H= Highly Associated
M= Moderately Associated
NPMI
max=
0.29
0.10
< 0

Annotator 2 \ Annotator 1		Anticipation		Joy		Trust		Fear		Surprise		Sadness		Disgust		Anger	
		H	M	H	M	H	M	H	M	H	M	H	M	H	M	H	M
Anticipation	H	0.13	0.04	0.04	0.05	0.02	0.04	-0.02	-0.01	-0.06	-0.03	-0.14	-0.09	-0.13	-0.11	-0.14	-0.11
	M	0.03	0.03	0.00	0.05	0.02	0.03	-0.03	-0.02	-0.03	-0.01	-0.06	-0.02	-0.08	-0.03	-0.06	-0.04
Joy	H	0.03	0.01	0.21	0.08	0.12	0.08	-0.23	-0.15	-0.04	-0.03	-0.27	-0.19	-0.23	-0.21	-0.27	-0.22
	M	0.04	0.03	0.07	0.13	0.05	0.08	-0.16	-0.06	-0.07	-0.03	-0.17	-0.11	-0.14	-0.12	-0.19	-0.13
Trust	H	0.03	0.02	0.10	0.08	0.13	0.08	-0.15	-0.09	-0.06	-0.04	-0.21	-0.12	-0.16	-0.14	-0.21	-0.15
	M	0.03	0.02	0.08	0.08	0.07	0.07	-0.12	-0.06	-0.04	-0.02	-0.13	-0.09	-0.14	-0.11	-0.17	-0.11
Fear	H	-0.02	-0.01	-0.23	-0.16	-0.15	-0.12	0.28	0.10	0.07	0.01	0.07	0.06	0.03	0.03	0.04	0.03
	M	0.01	-0.01	-0.15	-0.07	-0.10	-0.05	0.08	0.11	0.01	0.01	0.07	0.06	0.04	0.05	0.04	0.05
Surprise	H	-0.04	-0.02	-0.01	-0.05	-0.03	-0.05	0.05	0.01	0.15	0.03	-0.01	-0.01	0.01	-0.01	0.01	0.00
	M	-0.03	-0.01	-0.01	-0.02	-0.01	-0.01	0.01	0.01	0.02	0.03	0.00	0.02	0.00	0.01	-0.01	0.01
Sadness	H	-0.12	-0.06	-0.24	-0.18	-0.20	-0.13	0.07	0.06	0.02	0.02	0.27	0.13	0.11	0.11	0.10	0.12
	M	-0.06	-0.02	-0.18	-0.10	-0.13	-0.09	0.05	0.06	0.01	0.03	0.11	0.13	0.09	0.10	0.09	0.10
Disgust	H	-0.11	-0.05	-0.26	-0.17	-0.18	-0.15	0.05	0.05	0.05	0.03	0.10	0.10	0.21	0.13	0.18	0.12
	M	-0.10	-0.02	-0.19	-0.13	-0.14	-0.11	0.04	0.04	0.02	0.03	0.11	0.10	0.14	0.13	0.14	0.13
Anger	H	-0.12	-0.06	-0.25	-0.21	-0.20	-0.17	0.03	0.04	0.04	0.02	0.10	0.08	0.19	0.14	0.27	0.15
	M	-0.09	-0.03	-0.21	-0.15	-0.16	-0.11	0.04	0.05	0.02	0.03	0.12	0.11	0.10	0.14	0.13	0.15

Figure 7: NPMI confusion matrix on emotion categories for all annotator pairs with color scaling for legibility.

as for the REN (Henaff et al., 2017). We define a set of action function embeddings that can be applied to entities to change their state, **A**. After each sentence, the model selects an action function embedding to use to change the state of the entity memory. Unlike in Bosselut et al. (2018), these action function embeddings are not tied to real actions and are instead treated as latent t The dynamic memory is updated in the following way:

$$g_j = \sigma(s_t^T W_1[h_j, w_j]) \tag{15}$$

$$\alpha_t = softmax(MLP(s_t)) \tag{16}$$

$$a_t = \alpha_t^T \mathbf{A} \tag{17}$$

$$\tilde{h}_j = \phi(W_3 a_t + W_4 s_t) \tag{18}$$

$$h_j \leftarrow (1 - g_j)h_j + g_j \odot \tilde{h}_j \tag{19}$$

$$h_j \leftarrow \frac{h_j}{||h_j||} \tag{20}$$

where h_j is the value stored in memory cell j, w_j is a key corresponding to memory cell j, W_k are projection matrices, ϕ is a PReLU non-linearity and α_t is a distribution over possible action function embeddings. We initialize entity memory keys and entity memory values with the sum of the Glove vectors for all the words in the character name. Entity key values w_j are locked during training. We use dropout with p=0.5 between the encoder and dynamic memory.

C.5 LSTM Decoder

For the explanation generation task, we train a single-layer LSTM with a learning rate of 0.0003 and gradient clipping when the norm of the gradients exceeds 1. When outputting words, we concatenate the original hidden state from the encoder $\mathbf{h}^e$ to the output of the LSTM decoder before predicting a word. Word embeddings are initialized with pretrained Glove vectors (Pennington et al., 2014). In the generation task, the model must learn to generate individual annotations.

D Experimental Details

D.1 Data used for Annotation Classification Task

We split the development set into two parts, 80% used for training (D_1), 20% used for evaluating hyperparameters (D_2). We train a set of TF-IDF features for each word using all of the explanations from the real training set (D_t) and the portion of the development set used for training (D_1). We train a logistic regression classifier with L2 regularization. When training the classifier on D_1, we only train on examples where the explanation was written by a Turker who selected at least one Plutchik category label that was part of the majority set of Plutchik labels for the sentence the explanation and labels belong to. We prune D_2 and the test set similarly. We use individual annotations rather than majority labels to better learn specific fine-grained mappings.

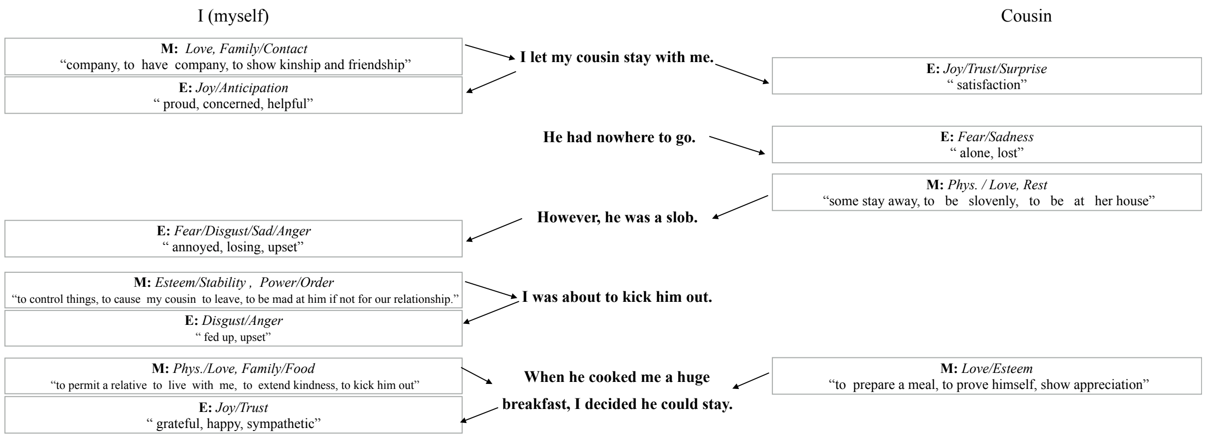
Emotional Explanation to Plutchik Labels We re-balance the dataset by sampling from the training set evenly among positive examples and negative examples for each category. We use L2 regularization with $\lambda = 0.1$ We include the full positive class distributions for each category in Table 6.

Motivation Explanation to Maslow Labels We use L2 regularization with $\lambda = 0.01$. The dataset is not rebalanced.

Motivation Explanation to Reiss Labels We use L2 regularization with $\lambda = 0.1$. The dataset is not rebalanced.

Maslow		Plutchik		Reiss			
Spiritual growth	6.24	Disgust	5.06	Status	2.53	Romance	2.00
Esteem	8.41	Surprise	11.99	Idealism	0.55	Savings	2.41
Love	11.35	Anger	5.78	Power	1.01	Contact	3.81
Stability	10.46	Trust	9.14	Family	3.57	Health	1.74
Physiological	4.37	Sadness	7.82	Food	2.87	Serenity	0.58
		Anticipation	16.65	Independence	1.29	Curiosity	2.58
		Joy	15.8	Belonging	0.14	Approval	1.88
		Fear	7.15	Competition	2.58	Rest	0.71
				Honor	0.67	Tranquility	2.34
						Order	2.56

Table 6: Class Distribution (percent positive instances) per category.


I (myself)

Cousin

M: *Love, Family/Contact*
"company, to have company, to show kinship and friendship"

E: *Joy/Anticipation*
" proud, concerned, helpful"

I let my cousin stay with me.

E: *Joy/Trust/Surprise*
" satisfaction"

He had nowhere to go.

E: *Fear/Sadness*
" alone, lost"

M: *Phys. / Love, Rest*
"some stay away, to be slovenly, to be at her house"

However, he was a slob.

E: *Fear/Disgust/Sad/Anger*
" annoyed, losing, upset"

M: *Esteem/Stability , Power/Order*
"to control things, to cause my cousin to leave, to be mad at him if not for our relationship."

E: *Disgust/Anger*
" fed up, upset"

I was about to kick him out.

M: *Phys./Love, Family/Food*
"to permit a relative to live with me, to extend kindness, to kick him out"

E: *Joy/Trust*
" grateful, happy, sympathetic"

When he cooked me a huge breakfast, I decided he could stay.

M: *Love/Esteem*
"to prepare a meal, to prove himself, show appreciation"

Figure 8: Fully annotated example from the annotation pipeline