Commonsense reasoning about social dynamics in text

Hannah Rashkin

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2020

Reading Committee:
Yejin Choi, Chair
Gina-Anne Levow
Noah A. Smith

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Commonsense reasoning about social dynamics in text

Hannah Rashkin

Chair of the Supervisory Committee:
Associate Professor Yejin Choi
Computer Science and Engineering

When humans interact with each other (e.g., having conversations, sharing stories, etc.), they are able to reason more deeply about social implications in order to better understand each other and have more productive interactions. For example, when hearing someone else discuss a personal story, most people are able to think about the consequences of various events, anticipate the feelings of their conversation partner, and respond accordingly.

Reasoning about social relationships in text is natural for most people, but is challenging for natural language processing models, in part because these relationships are often subtle, nuanced, and implicit. Training models for this type of inference is additionally challenging due to a lack of designated tasks, resources, and modelling frameworks specifically designed for this type of social commonsense reasoning.

We approach this problem by designing new focused tasks and resources specifically aimed towards types of social reasoning. We also introduce new modeling frameworks to learn to integrate social inferences with downstream tasks such as story and dialogue generation.

First, we investigate reasoning about social dynamics of characters and actions within stories. We create a new benchmark for reasoning about character mental state based on story events. We demonstrate that this type of reasoning is challenging even for state-of-the-art language understanding models. We also introduce plot dynamics as part of a new modeling framework for story generation. Our results indicate that tracking

plot state and integrating discourse features are beneficial for writing tighter narratives.

We also explore two types of reasoning about a speaker (e.g., a writer of a piece of text, a conversation partner, or so on) based on what they have said or written. We present connotation frames, a novel formalism for measuring connotative relationships implicit in the text that imply the writer's underlying message. We create a connotation frames lexicon, which may be useful in tasks like detecting implied stance, bias, or subtle meaning intended by a writer. Lastly, we investigate reasoning about a speaker in the dialogue setting by exploring the challenges of creating empathetic responses to a conversation partner. We introduce the task of empathetic response generation and a new dataset for training dialogue models to generate responses that are more empathetic and socially aware of a conversation partner's feelings.

Acknowledgements

I would like to thank my advisor, Yejin Choi, for all of her valuable guidance, feedback, and support throughout my PhD experiences. Additionally, I thank my other committee members: Noah Smith, Gina-Anne Levow, and Dan Weld. Thank you all for serving on my committee, offering mentorship and giving beneficial feedback on my research and thesis materials.

There are various other professors who have offered me guidance and advice throughout my graduate and undergraduate research careers. In particular, I want to thank Luke Zettlemoyer and Hannaneh Hajishirzi for giving me feedback on various projects. Additionally, I appreciate the support of Julia Hockenmaier, who acted as my advisor on undergraduate research projects. I also feel fortunate to have had multiple internships where I received very beneficial mentorship from my managers: Svitlana Volkova, Y-Lan Boureau, and Asli Celikyilmaz.

I also want to thank all of my various co-authors, collaborators, and workshop co-organizers including (in no particular order): Yejin Choi, Sameer Singh, Eunsol Choi, Luke Zettlemoyer, Maarten Sap, Marcella Cindy Prasettio, Ari Holtzman, Eric Bell, Svitlana Volkova, Emily Allaway, Noah Smith, Jin Yea Jang, Antoine Bosselut, Kevin Knight, Ronan Le Bras, Chandra Bhagavatula, Nicholas Lourie, Brendan Roof, Chaitanya Malaviya, Keisuke Sakaguchi, Doug Downey, Scott Wen-tau Yih, Derek Chen, Rowan Zellers, Yonatan Bisk, Ali Farhadi, Franziska Roesner, Asli Celikyilmaz, Marjan Ghazvininejad, Srini Iyer, Urvashi Khandelwal, Thomas Wolf, Eric Michael Smith, Margaret Li, Y-Lan Boureau, Jianfeng Gao, Michelle Ma, and Sophia Chan. I really am very lucky to have worked with you all!

I want to thank all of the fellow UW students who have supported and helped me. I want to especially thank the entire UW NLP student group who offered mentorship, feedback, and just plain moral support. I also appreciate the encouragement and positivity I've received from friends, both near and far - thank you

for always providing me with such a positive support system.

I would like to thank Elise deGoede Dorough, who has been so helpful throughout my PhD experience, answering questions and making suggestions about my degree program and various related logistics. Also, I owe a huge thank you to Chiemi Yamaoka, who has helped me so much with everything from ordering GPUs to managing crowdsourcing accounts to figuring out conference expenses. Thank you both – and to the rest of the UW CSE advising team as well – for making my PhD experience run so smoothly!

My graduate research was supported financially by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1256082 as well as grants and funding from DARPA and Samsung. Some of the work described in this thesis was also done while I was working as an intern at Microsoft or Facebook.

Lastly, of course, I would like to thank my family, who have all been incredibly supportive and helpful. To my parents and siblings, thank you for acting as great sounding boards. I always rely on your support and greatly appreciate it.

DEDICATION

To my parents

Contents

List of Figures

List of Tables

Chapter 1

Introduction

People are able to reason about social implications behind natural language. This skill allows them to make inferences about people being discussed in text or even about the writer themselves. For example, most people can easily identify the relationship between story characters' mental states and their behavior. This is a vital skill for both understanding and writing stories. This ability to reason also affects how humans interact with each other. For example, most people can empathize with a conversation partner and decide how to respond in a way that acknowledges their feelings.

As natural language processing (NLP) systems are increasingly used in interactive settings, social reasoning has become more important in end applications. For example, current dialogue systems such as those used in virtual assistants are currently only able to respond to direct commands. For instance, if a human user says "play me study music", a typical virtual assistant can respond appropriately (perhaps by offering a study playlist). However, if a use says something a bit more abstract ("I have a test tomorrow", "I'm planning to study tonight", etc.), most systems are unable to perform the multi-hop reasoning required to infer that it's still appropriate to offer to play study music. Beyond this single application, if an NLP system is reading or writing a story, as is common in many natural language understanding and generation tasks, deeper understanding of the characterization or plot trajectory requires being able to make inferences about the complex social dynamics between the characters, their actions, and other aspects of the plot.

Despite the significance of designing systems with these capabilities, there is a lack of NLP tasks specifically focused on investigating how well systems can infer social implications. We address this by introducing

several new tasks designed to investigate specific aspects of social commonsense reasoning related to the dynamics of people and their actions. We also present approaches for integrating this information in models for downstream tasks such as story generation or dialogue agents.

Previous approaches for modelling commonsense reasoning have looked to create knowledge graphs (Speer et al., 2017) or axiomatic rules (Gordon and Hobbs, 2017), but it is difficult or impossible to design a comprehensive set of rules or inferences. While machine learning approaches are a viable alternative, there is a lack of training resources in this area. Moreover, further analysis and carefully chosen examples (Trichelair et al., 2019; Davis and Marcus, 2015) shows that some machine learning models for commonsense tasks are better at detecting spurious correlations or lexical cues than performing generalizable reasoning. We approach these modelling challenges by designing resources for facilitating training and evaluating neural reasoning models. We also design new modelling frameworks that integrate social commonsense inference with NLP systems.

In this dissertation, we present novel work in bridging the gap between current NLP systems and human reasoning capabilities focused around social commonsense reasoning. First, we investigate modelling social dynamics within narratives by looking at relationships between mental states and actions. We also extend this to the problem of story generation where we design a new modelling framework that tracks plot states as a form of action dynamics. We also investigate what types of inferences can be made about a writer or speaker based on their utterances. We create a new formalism for connotative language which allows us to measure underlying writer intent behind a narrative. Lastly, we apply inferences about a speaker to the specific task of dialogue response generation where models may need to understand the mental state of a conversation partner to craft a better reply.

1.1 Background

1.1.1 Commonsense Reasoning

Commonsense reasoning has been a key goal of numerous artificial intelligence endeavors (Mccarthy, 1960; Gunning, 2018; Lenat, 1995; Davis and Marcus, 2015; Levesque, 2017). Many previous works have focused on creating benchmarks (Levesque, 2011; Sakaguchi et al., 2019; Roemmele et al., 2011; Sap et al., 2019b;

Talmor et al., 2019), knowledge graphs (Lenat, 1995; Speer et al., 2017; Sap et al., 2019a; Carlson et al., 2010), or axiomatic rules (Gordon and Hobbs, 2017). Recent progress in commonsense reasoning tasks has often used large neural networks. For example, BERT (Devlin et al., 2018) and GPT-2 (Radford et al., 2019) have been shown to be effective in various commonsense tasks. Similarly, Trinh and Le (2018) used large pre-trained language models to improve on the Winograd Schema challenge (Levesque, 2011). However, one limitation with neural approaches is that they typically require substantial training data. Furthermore, some recent work (Davis and Marcus, 2015; Trichelair et al., 2019) also shows that while large neural models are able to achieve high performance, they often rely on lexical correlations, and there is still room for improving on abstract, complex forms of reasoning.

While much of the previous work in commonsense reasoning has focused on physical or taxonomic knowledge (Speer et al., 2017; Lenat, 1995), we focus instead on *social commonsense reasoning*, which is the ability to infer deeper social implications beyond a surface level understanding. Social commonsense reasoning includes a broad set of reasoning about expectations involving human interactions and behavior. We focus on how social commonsense reasoning can be applied within natural language domains.

1.1.2 Social Dynamics

We specifically investigate social dynamics, which is a form of reasoning centered around the connection between people, their relationships with each other, their mental state, and how this relates to their actions and behavior.

Predicting Mental State

Many related works in NLP have focused on detecting mental state expressed in text, typically looking at detecting emotion (Duppada et al., 2018; Park et al., 2018; Xu et al., 2018b; Felbo et al., 2017; Mohammad and Turney, 2013) or predicting sentiment (Liu and Zhang, 2012; Wilson et al., 2005; Wiebe et al., 2005). Recently, Fu et al. (2019) also designed a new task for predicting intent behind advice, specifically focusing on the domain of advice-seekers on Reddit.

Reasoning about mental state also includes "theory of mind", an ability to reason about what others think or believe, which most people develop during childhood (Korkmaz, 2011; Blijd-Hoogewys et al., 2008;

Moore, 2013) and is an important component in how we communicate (Apperly, 2010). Nematzadeh et al. (2018) evaluated theory of mind reasoning skills in QA models, finding that keeping track of inconsistent beliefs of characters is challenging for models.

Another interesting dimension of modeling mental state is tracking how it changes over time. There is work in keeping track of entity state (Henaff et al., 2017), though it typically is aimed at more physical state changes such as in recipes (Kiddon et al., 2016; Bosselut et al., 2018). Many of these works have shown promising results that keeping track of these physical effects of actions is useful in natural language understanding and generation.

Predicting Action Dynamics

One long-standing goal in linguistics and NLP is to understand the dynamics of events. Work in identifying frames (Fillmore, 1982) has focused on creating sets of relations and attributes implied by an event, often based on social expectations. Script theory (Schank and Abelson, 1975) proposed the idea that there may be a typical sequence or grouping of expected actions that humans associate with a given event. For example going to a movie theater might involve first standing in line, buying tickets, and finding seats. Narrative cloze (Mostafazadeh et al., 2016) is a closely related task in which a story context is used to predict the most plausible next event. These types of action dynamics are closely related to social understanding because often they are defined by social expectations or the social consequences of certain actions. Work in narrative understanding and generation has made this connection by finding logical and coherent event chains, often through developing plot units (Lehnert, 1981; Goyal et al., 2010a) that relate character mental state to actions being taken.

1.1.3 Inferring Implicit Relationships

In this dissertation, we build new formalisms for understanding implicit social relationships found in text. Previous work in linguistics and NLP has covered a range of similar topics. For example, Lakoff (1993) proposed analysis of metaphors as a means of conveying implicit messages towards audiences. NLP models have been used to identify implicit subjectivity in text as a function of word choice (Greene and Resnik, 2009) or implicature (Wiebe and Deng, 2014). Work in detecting bias (Recasens et al., 2013; Fan et al.,

2019b; Da San Martino et al., 2019) has similarly designed methods for detecting subtle cues signifying the intent of the writer. Work in this area could have downstream consequences for designing approaches for automatically mitigating bias and identifying propaganda.

1.2 Challenges

One challenge in making social inferences is that they are usually subtly implied in text and therefore require "reading-between-the-lines". A related issue is the reporting bias problem (Gordon and Van Durme, 2013) in which only unusual occurrences are described in text. For example, writers rarely describe a character breathing because it is taken for granted that this must occur, but less common actions are mentioned more directly (e.g., according to Gordon and Van Durme (2013) murdering is described almost five times more than breathing in a large text corpus even though breathing is a much more ubiquitous action). These challenges make it difficult to find labelled data (or even heuristically-labelled data) for training a large-scale model.

Additionally, there is a lack of focused tasks available for social inference in the text domain. Many existing commonsense benchmarks (Talmor et al., 2019) and resources (Lenat, 1995; Speer et al., 2017) cover a much wider set of commonsense topics that include more physical or world knowledge. There are also existing NLP tasks where social understanding may be one component of evaluation (for example, a dialogue system may be judged in part on how engaging it is), but this is rarely the main focus of these tasks.

There is also a need for new models that can perform high-level reasoning. Large pretrained language models have made recent progress in commonsense reasoning tasks (Devlin et al., 2018; Trinh and Le, 2018; Radford et al., 2019), but it is difficult to interpret what they are learning. Error analysis (Sap et al., 2019b; Trichelair et al., 2019) frequently indicates these models rely heavily on lexical patterns rather than deeper reasoning mechanisms. In order for these models to be able to generalize to out-of-domain data or more challenging downstream tasks, they need to be enhanced with more high-level reasoning capabilities about these types of interactions.

Lastly, another key challenge is designing evaluation metrics for models with commonsense reasoning. Social reasoning usually involves distributional inferences, or rather a prediction about what's most likely amongst multiple plausible inferences (de Marneffe et al., 2012). For example, when thinking about what

might occur next in a story, there may be multiple plausible next actions with some more likely than others. Interpretation may also vary depending on culture, personal experiences, and additional context about the situation. For these reasons, label-based classification accuracy is difficult to use as a metric. For evaluating open-ended inferences, automatic metrics from text generation literature also has clear limitations. Because there may be more than one plausible output, many word-overlap-based generation metrics are not the most appropriate measures of inference accuracy. Additionally, previous work (e.g., Liu et al. (2016)) has shown that often these automatic generations metrics do not appropriately correlate with human judgments. Human evaluations may be the closest available proxy for understanding how these systems would work in "real world" applications. However, these can often be costly to scale up. In this thesis, we approach this challenge by including a wide array of metrics for each task as a way of characterizing multiple aspects of model behavior. We generally focus on human evaluations as a method for drawing conclusions but also explore using automatic metrics for drawing additional insights.

1.3 Approach

We address these challenges in a multi-faceted approach for studying social dynamics in text. First, we design focused tasks for evaluating how well NLP systems can perform different aspects of social commonsense reasoning. We also create new resources for facilitating training and evaluation. Lastly, we present new frameworks for integrating understanding of implicit social dynamics with ML models. Due to the limitations of automatically evaluating new models, we also draw insights using human judgments obtained through crowdsourcing.

In this dissertation, we investigate four tasks described in Figure 1.1. These tasks can be grouped into two themes: First, we explore types of inferences that can be made about the social dynamics of story characters and plot events. Then, we also investigate challenges related to making inferences about a speaker, a key part of any collaborative or interactive system.

In Chapter 2, we investigate how well NLP models can understand the mental state of characters based on story events. We next explore ways of tracking plot dynamics while generating new narratives based on plot outlines (Chapter 3). Then, in Chapter 4, we design a new formalism for inferring relationships connoted by a writer. Finally, Chapter 5, we use inferences about the mental state of a speaker to craft

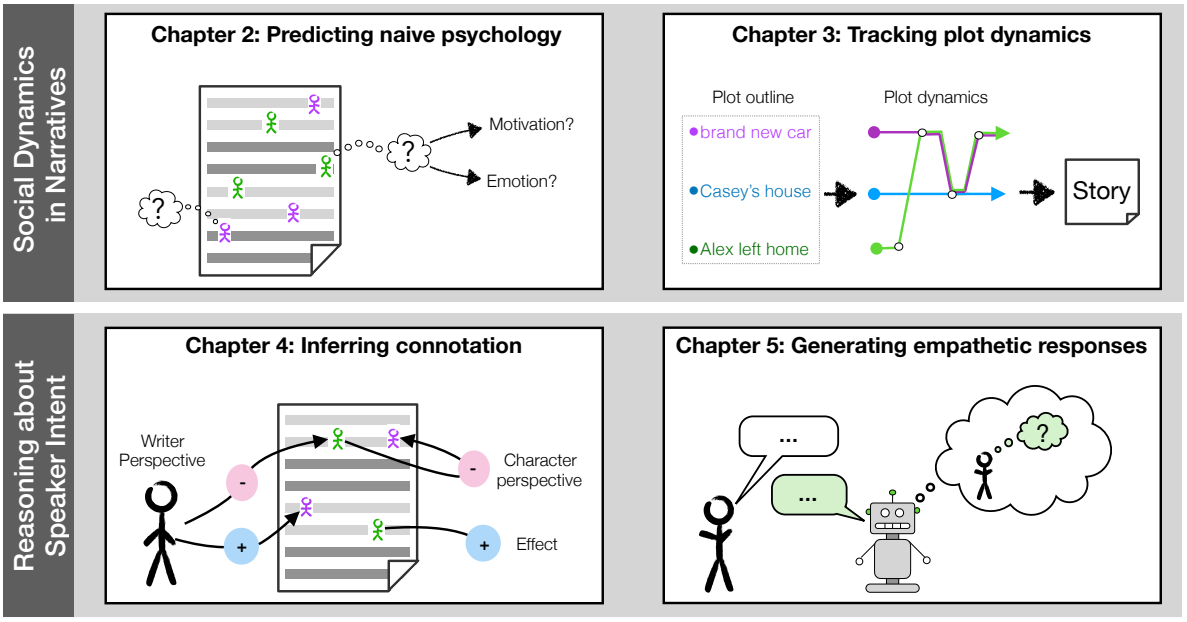


Figure 1.1: Overview of approach: In this dissertation, we cover four projects related to social commonsense reasoning.

empathetic responses in a dialogue setting.

1.4 Outline

In Chapter 2, we present a novel dataset for training and evaluating models' inferences of the naive psychology of story characters. This new resource includes annotations for each character's motivation and emotional reaction changes over the course of 15k short commonsense stories. We establish baseline performance of machine learning models on two different tasks for predicting mental state changes of characters and demonstrate that this is a significant challenge for NLP systems.

Next, in Chapter 3, we investigate how to write stories by threading plot events together in a cohesive way. We present a new transformer model that encodes plot dynamics with state tracking and discourse-based features. Through comprehensive experiments and analysis, we compare performance of our model with competitive baselines at writing stories based on plot outlines, demonstrating that our model generally writes tighter narratives.

In Chapter 4, we focus on more implicit relationships, merely connoted by events and interactions described in text. We focus on how the choice of verb used to describe an event affects the way in which

people are portrayed to the audience. We present connotation frames, a new formalism representing four new types of relationships connoted by verb choice. We annotate a lexicon of verbs and show that machine learning models can also be used to extend the lexicon. The resulting lexicon can be used for downstream analysis tasks. We describe analysing implicit stances discussed in a large news corpus as a proof-of-concept.

Lastly, in Chapter 5, we present empathetic response generation, a new task for evaluating how well dialogue systems respond to personal stories being shared by a conversation partner. We create a new dataset of 25k conversations conditioned on emotion prompts. We evaluate how dialogue models can leverage this data to craft more empathetic replies.

Chapter 2

Reasoning about Naive Psychology in Narratives

Humans are able to reason about story characters as they read about them. For example, they can infer how characters feel over the course of the story and use this information to predict future actions. In this chapter, we present a new dataset for studying the naive psychology of characters in stories. We collect annotations of motivations causing story events as well as emotional reactions that are caused by story events. We also establish baseline performance of machine learning models at the task of predicting mental state. We demonstrate that this is a challenging open problem for NLP systems with a few promising directions for future research. This chapter is based on work originally published in Rashkin et al. (2018a).

2.1 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.

To address these challenges, 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 Story Commonsense dataset offers three unique properties. First, as highlighted in Figure 2.1, 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 2.1, 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" $\rightarrow$ Maslow's category *love*). Our corpus spans across 15k stories, amounting to 300k low-level annotations for around 150k character-line pairs.

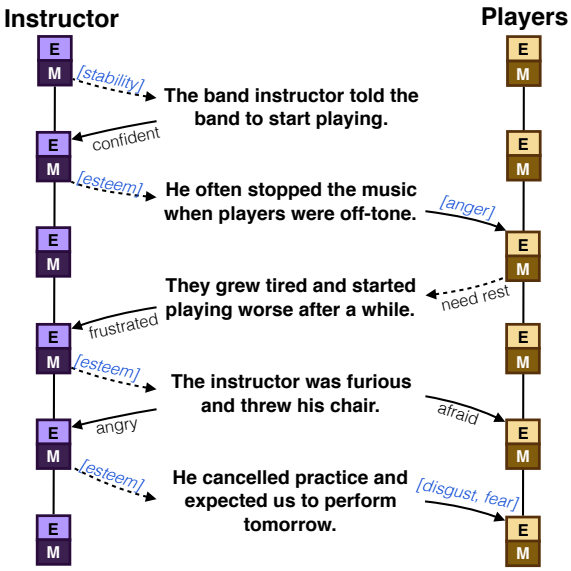


Figure 2.1: A story example with partial annotations for motivations (dashed) and emotional reactions (solid). Open text explanations are in black (e.g., "frustrated") and formal theory labels are in blue with brackets (e.g., "[esteem]").

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.

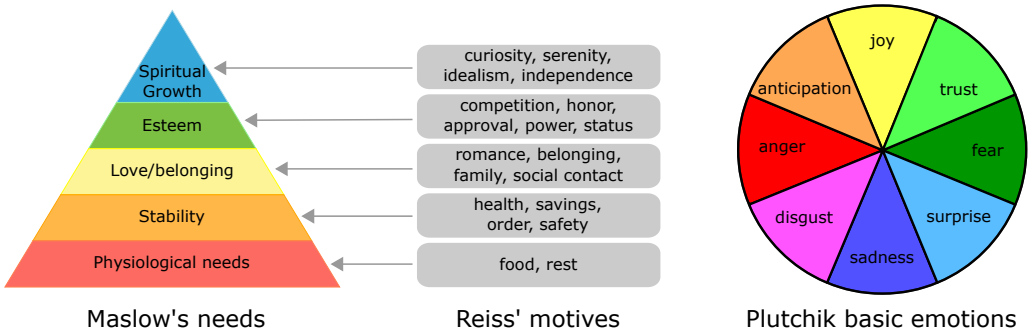


Figure 2.2: Theories of Motivation (Maslow and Reiss) and Emotional Reaction (Plutchik).

2.2 Related Work

Naive Psychology in Stories Previous work in plot units (Lehnert, 1981) developed formalisms for affect and mental state in story narratives that included motivations and reactions. Inspired by this body of work, we collect mental state annotations for stories to be used as a new resource in this space.

Similarly, there have been multiple works 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). We use the commonsense stories from Mostafazadeh et al. (2016) as a resource for annotating both emotions and motivations for multiple characters.

Modeling Entity State 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 in multiple tasks while providing some interpretability for predictions. Tracking entity state is useful for our task, in which we track emotional reactions and motivations of characters in stories. We use two of these entity models (Henaff et al., 2017; Bosselut et al., 2018) for investigating how much performance can be improved with explicit entity representations.

2.3 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). In addition, they depict multiple interactions between story characters,

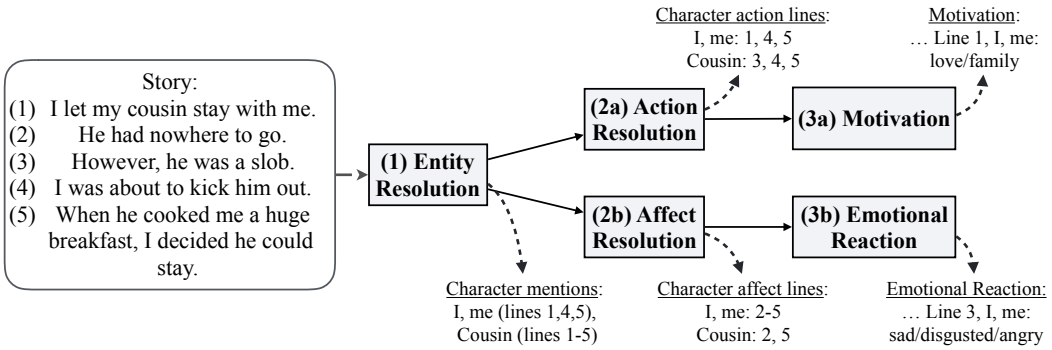


Figure 2.3: The annotation pipeline for the fine-grained annotations with an example story.

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.

In addition to collecting open-text explanations of characters' motivations and emotions, we use psychology categories from theories of motivation (using Maslow categories with Reiss subcategories (Maslow, 1943; Reiss, 2004)) and emotion (using Plutchik categories (Plutchik, 1980)) as shown in Figure 2.2.

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., 2010a; 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 2.3. All annotations were collected using crowdsourced workers from Amazon Mechanical Turk.

2.4 Annotation Pipeline

We describe the components and workflow of the full annotation pipeline shown in Figure 2.3. The example story in the figure is used to illustrate the output of various steps in the pipeline.

(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 2.3, the characters identified by annotators are "I/me" and "My cousin", who 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 sentences 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 2.3, 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 annotations 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

		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 2.1: Annotated data statistics for each dataset

subcategories. In Figure 2.3, 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 2.3, 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*.

2.5 Dataset Statistics and Insights

Cost The tasks corresponding to the theory category assignments are the hardest and most expensive in the pipeline ($\sim$\$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 ($\sim$\$1 per story).

Scale Our dataset 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 150k 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 2.1 shows the breakdown across training, development, and test splits.

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

Agreements For the categorization sets (Maslow, Reiss and Plutchik), we compare the performance of annotators by treating each individual category as a binary label 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 2.2. We also compute the percent agreement between the individual annotations and the majority labels.[1]

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 2.4, 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

[1]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.

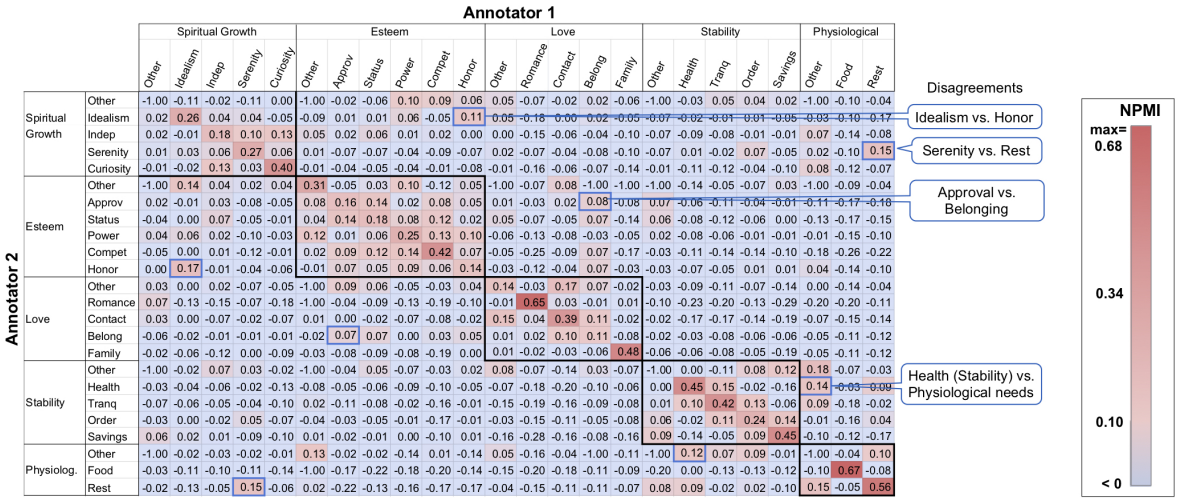


Figure 2.4: NPMI confusion matrix on motivational categories for all annotator pairs with color scaling for legibility. The highest values are generally along diagonal or within Maslow categories (outlined in black). We highlight a few common points of disagreement between thematically similar categories.

most confusions occurring between Reiss motives in the same Maslow category (outlined black in Figure 2.4). Other disagreements generally involve Reiss subcategories that are thematically similar, such as *serenity* (mental relaxation) and *rest* (physical relaxation). We also perform this analysis for Plutchik categories, 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*).

2.6 Tasks

The multiple modes covered by the annotations in this new dataset allow for many new tasks to be explored. We outline two task types below.

State Classification The primary set of tasks involves categorizing the psychological states of story characters, using one of the label sets (Plutchik, Maslow, or Reiss) collected for the dev and test splits of our dataset. A model is given a story character and a line from the story (along with optional preceding context lines) and predicts the motivation or emotional reaction of the character in that line. As discussed in the challenges section of Chapter 1, one issue in evaluating social commonsense reasoning is that there may be multiple plausible inferences that can be made. To address this, we frame this task as a multi-label problem. A binary label is predicted for each of the Maslow needs, Reiss motives or Plutchik categories, respectively.

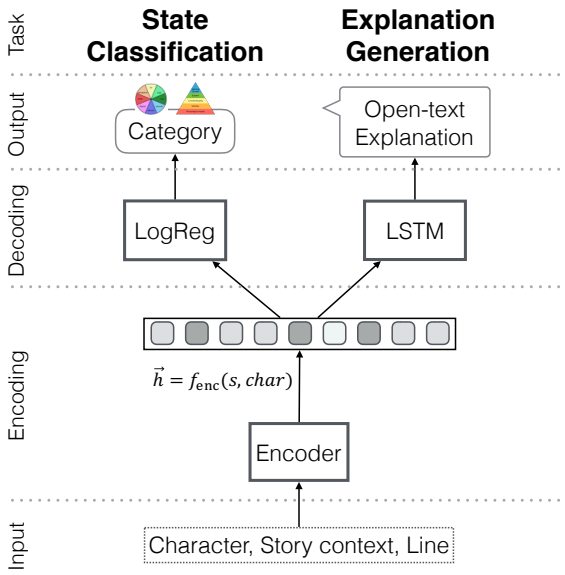


Figure 2.5: General model architectures for two new task types

Explanation Generation We can use the open text explanations to train models to describe the psychological state of a character in free text. These explanations allow for two conditional generation tasks where the model must generate the words describing the emotional reaction or motivation of the character.

2.7 Baseline Models

The general model architectures for the two tasks are shown in Figure 2.5. We describe each model component below. Because the two task use similar encoders, models could be trained separately or in a multi-task set-up.

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:

$$\mathbf{h}^e = \text{Encoder}(\mathbf{x}^s, \mathbf{C}[e_j]) \tag{2.1}$$

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

2.7.1 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 (all the preceding sentences mentioning the character, $\mathbf{C}[e_j] \rightarrow \mathbf{h}^c$) 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 $\mathbf{h}^s$:

$$\mathbf{h}^s = \phi(W_s v^s + b_s) \tag{2.2}$$

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 2.2).

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 $w_n^s \in \mathbf{x}^s$.

$$v^s = [x_0^s, x_1^s, \ldots, x_N^s] \tag{2.3}$$

$$\mathbf{h}^s = \text{CNN}(v^s) \tag{2.4}$$

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.

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:

$$\mathbf{h}^e = \{m_j\}^s \tag{2.5}$$

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

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.

2.7.2 Decoders

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:

$$\hat{y}_z = \sigma(W_z \mathbf{h}^e + b_z) \tag{2.6}$$

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

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).

2.7.3 Training Set-up

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:

$$\mathcal{L} = \sum_{z=1}^{Z} \gamma_z y_z \log \hat{y}_z + (1 - \gamma_z)(1 - y_z) \log(1 - \hat{y}_z) \tag{2.7}$$

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

$$\gamma_z = 1 - e^{-\sqrt{P(y_z)}} \tag{2.8}$$

where $P(y_z)$ is the marginal class probability of a positive label for z in the training set. The γ terms allow for soft re-weighting of the unbalanced class labels.

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:

$$L_{gen} = -\sum_{t=1}^{T} \log P(y_t | y_0, ..., y_{t-1}, \mathbf{h}^e) \tag{2.9}$$

where $\mathbf{h}^e$ is the sentence-character representation produced by an encoder from Section 2.7.1.

2.8 Experiments

2.8.1 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 2.3. Note that random is low because the distribution of positive instances for each 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 As discussed in the Chapter 1, automatic evaluations for open-ended generations are limited and often poorly correlated to human judgment. In this case, because explanations tend to be short sequences with high levels of synonymy, traditional metrics such as BLEU are particularly inadequate for evaluating generation quality. Instead of using overlap-based metrics, we investigate performance using vector-based metrics that are better suited for dealing with synonymous or short answers. We use the vector average and vector extrema metrics from Liu et al. (2016) using GloVe vectors (Pennington et al., 2014) of generated and reference words. In the vector average metric, we compare the cosine similarity of the average embedding of the words in the gold response with the average embedding of the words in the generated output. The vector extrema measure is calculated similarly with element-wise max operations in place of averaging. While this metric is not an ideal replacement for human judgment, it allows us to draw some insights about whether the output is semantically similar to the gold responses. We report results in Table 2.4 on the dev. set and compare to a baseline that randomly samples an example from the dev. set as a generated sequence.

2.8.2 Ablations

Character-specific 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 2.3, this ablation is the first row for each encoder presented.

Explanation Pretraining Because the state classification and explanation generation tasks use the same types of encoders, we explore initializing a classification encoder with parameters trained on the generation task. This may be useful because the large-scale training set only has free text explanations. We pretrain the CNN, LSTM, REN, and NPN encoders using the explanation generation task to produce emotion or motivation explanations with the training set. We use the parameters from this model to initialize the encoders when training for the state classification task. In Table 2.3, we denote this extra pre-training step as the models "+ explanation training".

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 2.3: State Classification Results

2.8.3 Experimental Results

State Classification We show results on the test set for categorizing Maslow, Reiss, and Plutchik states in Table 2.3. 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 an example 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. This analysis indicates that robust context representations may be an important avenue for future research in this task.

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 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. This may indicate that there are lexical correlations that these models are learning, rather than more abstract forms of reasoning. One reason why joy may be easier to predict is that it is a more frequently appearing class. One direction for future research may be to incorporate external resources (such as external knowledge graphs (Sap et al., 2019a), in order to increase performance on rarer classes like disgust.

Explanation Generation We provide results for the task of generating explanations of motivations and emotions in Table 2.4 using the vector-based evaluation metrics described in Sec. 2.8.1. Because the explanations are generally emotion or motivation words, the randomly selected explanation can often be close in embedding space to the reference explanations, making the random baseline fairly competitive. This may be one limitation of using vector-based evaluations that future work can improve on. However, even with the limitations of vector-based evaluations, it is evident that all models outperform the random baseline on both metrics, indicating that the generated explanations are closer semantically to the reference annotation. Results indicate that models with entity state tracking may be slightly better at open-ended explanation generation, so this may be a promising direction for future work.

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 2.4: Vector average and extrema scores for generation of annotation explanations

2.9 Summary

In this chapter, 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 establish baseline performance on this new resource. Importantly, we show that modeling character-specific context and pretraining on free-response data can

boost labeling performance. Results demonstrate that this is still a challenging open task. Future work may be able to improve performance by investigating more robust context representations, including external resources, or exploring more abstract multi-hop reasoning.

Chapter 3

Learning to Track Plot Dynamics in Story Generation

In the previous chapter, we discussed inferences about the dynamics of mental states based on character interactions in stories. In this chapter, we delve further into the dynamics of story events. We focus on how plot elements get interwoven together in the context of writing a coherent narrative. For humans, the story writing process typically begins with an outline containing various plot points that need to be included. For translating these plot outlines to stories, human writers must think about the most plausible ordering and how to combine them in a coherent way to tell a tighter story, requiring commonsense reasoning about social dynamics. We simulate this process by designing a novel transformer model that dynamically tracks plot state while writing long-form narratives. This chapter is based on work from Rashkin et al. (2020).

3.1 Introduction

Composing a story requires a complex planning process. First, the writer starts with a sketch of what key characters and events the story will be roughly about. Then, as they unfold the story, the writer must keep track of the elaborate plot that weaves together the characters and events in a coherent and consistent narrative.

We study this complex storytelling process by formulating it as the task of *outline-conditioned story*

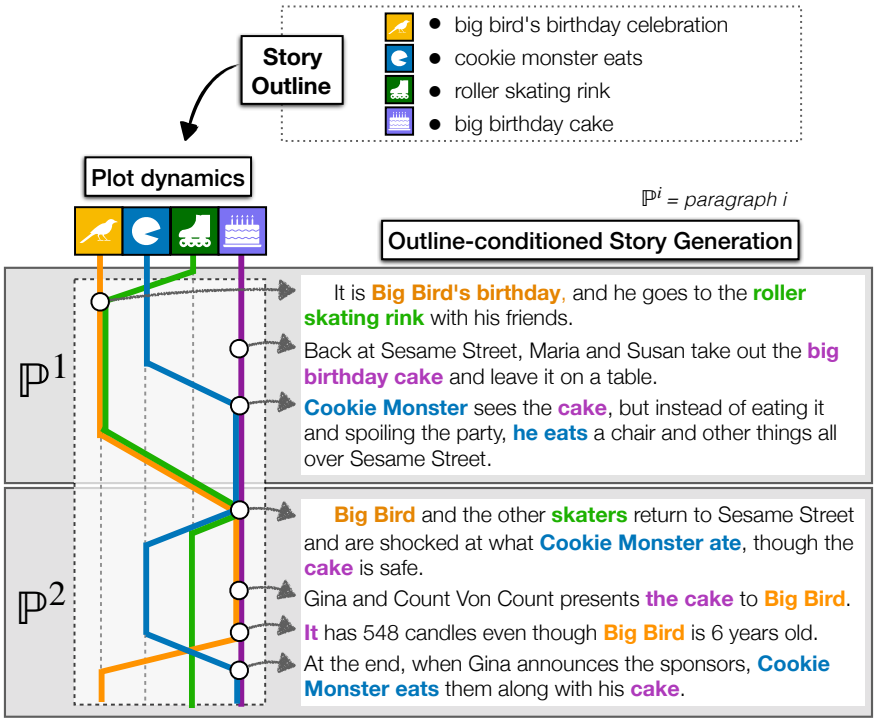


Figure 3.1: An outline (input) paired with a story (output) from our Wikiplot training set. Plot elements from the outline can appear and reappear non-linearly throughout the plot, as shown in plot dynamics graph. Composing stories from an outline requires keeping track of how outline phrases have been used while writing.

generation, illustrated in Figure 3.1. Given an outline, a set of phrases describing key characters and events to appear in a story, the task is to generate a coherent narrative that is consistent with the provided outline. This task is challenging as the input provides only the rough elements of the plot[1]. Thus, the model needs to flesh out how these plot elements will intertwine with each other across different parts of the story. The flowchart in Figure 3.1 demonstrates an example of a latent plot structure: different key phrases from the outline appear and re-appear jointly throughout different sentences and paragraphs. Notably, the way that outline points are interwoven needs to be determined dynamically based on what's already been composed while also staying true to the original outline and overall narrative structure.

We present PLOTMACHINES, a novel narrative transformer that simulates the outline-conditioned generation process described above. Our model learns to transform an outline into a multi-paragraph story using dynamic memory blocks that keep track of the implicit plot states computed using the outline and

[1]Here, we define plot as the main sequence of events in the story.

the story generated thus far. We draw inspiration from prior work in dialogue state tracking (Thomson and Young, 2010; Lee, 2013; Chao and Lane, 2019), entity tracking (Henaff et al., 2017; Bosselut et al., 2018), and memory networks (Sukhbaatar et al., 2015) for keeping track of plot states. We also inform our model with high-level narrative structure using discourse labels so that it can learn different styles of writing corresponding to different parts of the narrative (i.e. beginning, middle, and end). PLOTMACHINES is, to the best of our knowledge, the first model designed to generate multi-paragraph stories conditioned on outlines and can be trained end-to-end to learn the latent plot patterns without explicit plot annotations for supervision.

To support research on outline-conditioned generation, we present three new datasets, including both fiction and non-fiction domains, where multi-paragraph narratives are paired with automatically constructed outlines using state-of-the-art key phrase extraction. Importantly, our task formulation of outline-conditioned generation is general and can be applied to various forms of grounded language generation. Comprehensive experiments on these datasets demonstrate that recently introduced state-of-the-art large-scale language models such as GPT-2 (Radford et al., 2019) and GROVER (Zellers et al., 2019), despite their impressive generation performance, still struggle to generate coherent narratives that are consistent with input outlines. Our experiments indicate that dynamic plot state tracking is important for constructing narratives with tighter and more consistent plots compared to competitive baselines.

3.2 Related Work

State Tracking There is a plethora of work in state tracking for dialogue where memory states are updated after each utterance (Thomson and Young, 2010; Young et al., 2010; Lee, 2013; Chao and Lane, 2019). Similarly, SC-LSTMs (Wen et al., 2015) dynamically updated dialogue act representations as a form of sentence planning in spoken dialogue generation. Memory and entity networks (Henaff et al., 2017; Sukhbaatar et al., 2015) and neural checklists (Kiddon et al., 2016) also have used similar methods for tracking entities for other tasks. In this work, we adapt these techniques for generating stories where we keep track of plot state that is updated after each paragraph. Our method of decoding paragraphs recurrently also draws on existing work in hierarchical decoding (Li et al., 2015; Shen et al., 2019), which similarly decodes in multiple levels of abstraction over paragraphs, sentences, and words.

Controllable Story Generation There have been a variety of works focusing on generating stories in plot-controllable or plan-driven ways (e.g. (Riedl and Young, 2010; Fan et al., 2018; Peng et al., 2018; Jain et al., 2017)). Outline-conditioned generation is complementary to these tasks in that outlines provide more flexibility than very fine-grained srl-based or event-based plans (Fan et al., 2019a; Martin et al., 2017; Harrison et al., 2017) and more grounding than coarse-grained prompts (Fan et al., 2018; Xu et al., 2018a) or five-word storylines (Peng et al., 2018; Yao et al., 2019). Similar to many recent works in this area, we focus on seq2seq-based approaches, which we implement using transformers. We further expand upon the modeling for the challenges specific to our task using our approaches for state tracking and applying discourse structure.

3.3 Task: Outline-Conditioned Generation

We introduce the task of outline-conditioned story generation, which takes a plot outline as input and produces a long, multi-paragraph story. Similar to planning-based story writing formulations (Porteous and Cavazza, 2009; Riedl, 2009; Riedl and Young, 2010; Zhou et al., 2018a; Fan et al., 2019a; Pérez y Pérez and Sharples, 2001), outline-conditioned generation aims to closely mirror the way humans typically write long documents by interweaving key points.

We formulate the outline as a list of un-ordered bullet points which reflect key words, phrases, or sentences to be integrated in the output narrative. These plot outlines are inspired, in part, by previous work in short-form story generation tasks that conditioned on storylines (Peng et al., 2018; Yao et al., 2019), which were defined as an ordered list of exactly five single-word points. We extend this concept to long-form story generation by defining a plot outline more flexibly as: an *un-ordered* list of *an arbitrary number of multi-word* plot elements. An outline also differs from a writing prompt, such as those found in other controllable writing tasks (Fan et al., 2018), which are typically more abstract and often just a starting point for a story. Unlike a prompt, an outline is a list of concrete points that must appear somewhere in the story content.

One challenge in this task is to create stories that have appropriate discourse that are not overly repetitive, follow a logical order, and have realistic beginnings and endings. A second challenge is for stories to include the outline in a natural way. For example, it may be appropriate for certain outline points to be mentioned later on in the story (e.g. the protagonist dying may be more typically at the end of the story).

Wikiplots # stories : 130k avg # pars : 3.1 data-split : 90/5/5	**Outline**: • the rocky horror picture show • convention attendees includes servants (...) **Story:** A criminologist narrates the tale of the newly engaged couple, Brad Majors and Janet Weiss, who find themselves lost and with a flat tire on a cold and rainy late November evening, somewhere near Denton in 1974 (...)
WritingPrompts # stories : 300k avg # pars : 5.9 data-split : 90/5/5	**Outline**: • found something protruding • geometric shapes glowing • sister kneeling beside • dead bodies everywhere • darkness overwhelmed • firelight flickering (...) **Story**: It was dark and Levi was pretty sure he was lying on his back . There was firelight flickering off of what was left of a ceiling . He could hear something but it was muffled . He (...)
NYTimes # stories : 240k avg # pars : 15.2 data-split : 90/5/5	**Outline**: • upcoming annual economic summit meeting • take intermediate steps (...) **Article:** The long-simmering tensions in Serbia's province of Kosovo turned violent in recent weeks and threaten to ignite a wider war in the Balkans. Only a concerted diplomatic effort by the United States can keep the conflict from escalating. Though he has been attentive to the problem (...)

Table 3.1: Datasets used in the experiments showing the number of stories, the average number of paragraphs per story, and the split of stories across train/dev/test. We also show excerpts from examples of outlines paired with a story.

3.4 Data: Outline to Story

We construct three datasets for outline-conditioned generation. We focus on fictitious generation, but also include the news domain for generalization. We build on existing story datasets for the target narratives, which we pair with automatically constructed input outlines as described below. We show examples from each dataset in Table 3.1.

Wikiplots The Wikiplots corpus[2] consists of plots of movies, TV shows, and books scraped from Wikipedia. For our task, we divide stories from the corpus into 90% train, 5% validation and 5% test splits.

WritingPrompts WritingPrompts (Fan et al., 2018) is a story generation dataset, collected from the /r/WritingPrompts subreddit – a forum where Reddit users compose short stories inspired by other users' prompts. It contains over 300k human-written (prompt, story) pairs. We use the same train/dev/test split from the original dataset paper.

NYTimes Unlike the other two datasets, NYTimes (Sandhaus, 2008) contains news articles rather than fictional stories. We use this dateset to investigate how well our approach generalizes to nonfiction. Due to concerns over fake news creation (Zellers et al., 2019), we only use this dataset for quantitative exploration and will not publicly release the models.

[2] https://github.com/markriedl/WikiPlots

Outline Extraction Because these datasets do not already have plot outlines to use as input, we extract lists of plot points using the RAKE (Rapid Automatic Keyword Extraction) algorithm (Rose et al., 2010)[3]. RAKE is a domain independent keyword extraction algorithm, which determines key phrases in a document by analyzing the word frequency and co-occurrence with other words in the text. We filtered key-points with overlapping n-grams. This is inspired by similar RAKE-based methods for creating storylines (Peng et al., 2018), but differs in that we extract longer plot points (each of which is 3-8 words) and do not order the points within the outline. For WritingPrompts and NYTimes, we also use sentences from the prompt and article abstract as additional sources for outline points.

3.5 PLOTMACHINES

Our approach to this task is to design a model that combines recent success in text generation with transformer-based architectures (Vaswani et al., 2017) with memory mechanisms that keep track of the plot elements from the outline as they are used in the story. We also incorporate special discourse features into the modelling to learn a structure over the long multi-paragraph story format.

We introduce PLOTMACHINES (PM), an end-to-end trainable transformer built on top of the GPT model[4] (Radford et al., 2018), as shown in Figure 3.2. Given an outline as input, the model generates paragraphs, recurrently, while updating a memory matrix M that keeps track of plot elements from the outline. This paragraph-level generation framework is motivated by human writing styles, in which each paragraph is a distinct section of related sentences.

At each time step, i, PLOTMACHINES generates a new paragraph $\mathbb{P}^i$:

$$(\mathbb{P}^i, h^i, M^i) = \text{PM}(o, d^i, h^{i-1}, M^{i-1}) \tag{3.1}$$

where o is the input outline representation (described in Sec. 3.5.1), d^i is the discourse representation associated with paragraph i (Sec. 3.5.2), h^{i-1} is a vector representation of the preceding story context (Sec. 3.5.3), and M^{i-1} is the previous memory (Sec. 3.5.4).

[3] https://pypi.org/project/rake-nltk/

[4] We build on top of GPT, though our approach could be used with most transformer-based LMs. In experiments, we also look at a version of PLOTMACHINES using GPT-2 (Radford et al., 2019) as a base.

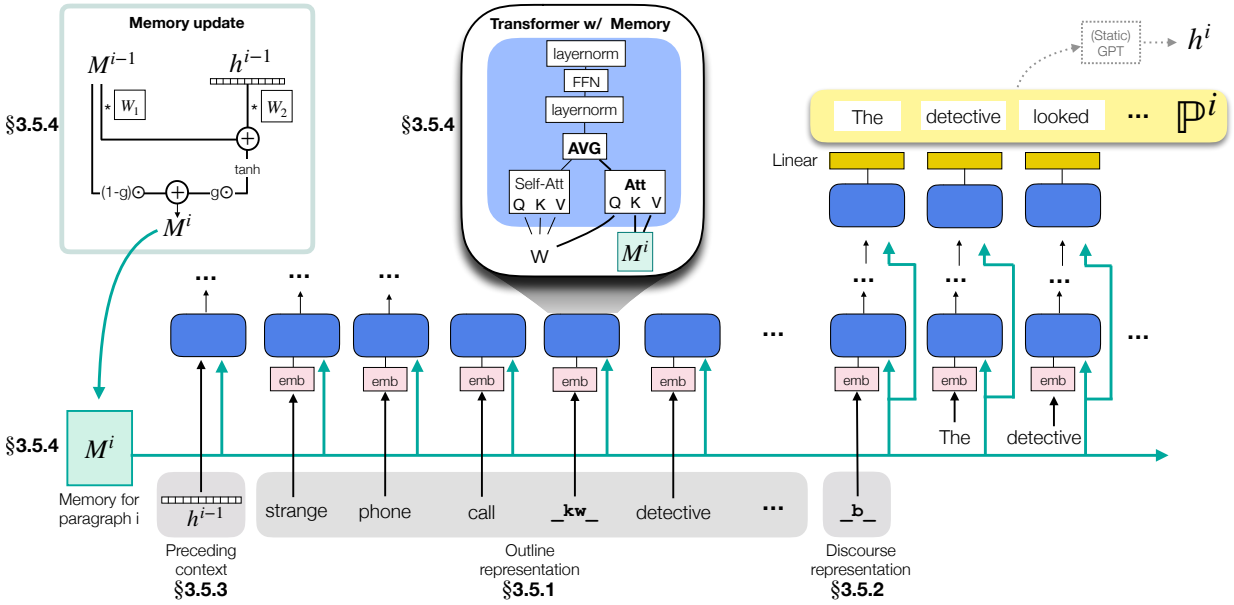


Figure 3.2: PLOTMACHINES: The model generates a paragraph $\mathbb{P}^i$ using the memory (M^{i-1}), the previous paragraph representation (h^{i-1}), the outline representation (o) and discourse representation (d^i). First, a gated update mechanism updates the memory using the previous memory and previous paragraph representation. Each transformer block includes an extra attention over the current memory matrix M^i. The previous paragraph representation, h^{i-1}, the outline, and discourse tag (e.g. _b_) are also prepended to the generation as an input sequence (grayed inputs). The output tokens of the generated paragraph are used to compute h^i using a static GPT model.

3.5.1 Outline Representation

The plot outline (i.e. the input to the model) is treated as a sequence of tokens, o, and used as input for the transformer. We use special **_kw_** tokens to delimit each plot point in the outline. The end of the outline is marked with a special **_endkw_** token. We truncate the entire outline to maximum of n tokens. For example, an outline containing two plot points ({'strange phone call', 'detective'}) is turned into the input sequence:

`strange phone call` **`_kw_`** `detective` **`_endkw_`**

3.5.2 Discourse Representation

We posit that there are stylistic differences in how the beginning, middle and end of a story are written. To maintain these differences, we introduce d^i, discourse information about whether the i-th paragraph is an

introduction, body, or conclusion paragraph. We add special tokens **_i_**, **_b_**, **_c_** for the introduction, body, and conclusion paragraphs respectively[5]. The appropriate discourse token is appended to the outline representation as part of the input sequence.

3.5.3 Preceding Context Representation

With the goal of incorporating previous story context in generating each paragraph, we use h^{i-1}, an embedded representation of the previous paragraph, which is added to the model input. More concretely, h^{i-1} is computed as the average embedding of output representations of the previous paragraph words (using a static GPT model). The h^{i-1} vector is used as an initial input to the transformer architecture, as shown in Figure 3.2.

3.5.4 Memory Representation

We implement a memory in two parts to address two challenges in outline-conditioned generation. First, we want to keep track of the portions of outline that have been mentioned. Second, we want to maintain semantic coherence across the entire story. To address these two challenges, we implement the memory as consisting of two parts: K, a set of vectors keeping track of outline points, and D, a matrix that stores a latent topic distribution of what's been written so far.

Notation: We define d as the embedding size of the transformer model and n as the maximum number of tokens in the outline. Memory is treated as a $\mathbb{R}^{d \times 2n}$ matrix which consists of two smaller matrices stacked together ($M = [K; D]$). K is a $\mathbb{R}^{d \times n}$ representation of outline points and D is a $\mathbb{R}^{d \times n}$ representation of the latent document state. K is initialized with embeddings representing each of the tokens in the outline and D is randomly initialized. The j-th column of memory at the timestep for paragraph i will be denoted M_j^i.

Updating Memory: The memory is updated (top left corner of Fig. 3.2) using h^{i-1}, the average GPT output representation of the previous paragraph. We use update equations based on those in entity-based models such as Henaff et al. (2017). Importantly, we use a gating mechanism, g, to allow the model to learn to

[5]We make the simplifying assumption that the first paragraph is an introduction, the last paragraph is the conclusion paragraph, and the other paragraphs are all body paragraphs.

flexibly control how much each cell in memory is updated based on what's changed in the story, as below:

$$\hat{M}_j^i = tanh(W_1 M_j^{i-1} + W_2 h^{i-1}) \tag{3.2}$$

$$g_j^i = sigm(W_3 M_j^{i-1} + W_4 h^{i-1}) \tag{3.3}$$

$$M_j^i = (1 - g_j^i) \odot M_j^{i-1} + g_j^i \odot \hat{M}_j^i \tag{3.4}$$

where all W's are matrices of dimension $\mathbb{R}^{d \times d}$.

Transformer Blocks with Memory: Lastly, we must alter the GPT transformer blocks to include the memory in the language modeling. We augment the transformer blocks to contain two parallel attention modules, as shown in Figure 3.2. One attention module (on the left in the figure) performs the standard GPT self-attention using transformer inputs to create queries, keys, and values. The other attention module uses transformer input to attend over the memory vectors (i.e., using the memory for creating key and value vectors). The outputs of both attention modules are averaged[6] before performing the additive layer-norm.

3.5.5 Training and Decoding

At training time, the model is trained end-to-end on the cross-entropy loss of predicting each paragraph. Gold representations of previous paragraphs in the story are used to update the memory and compute h^{i-1}. At decoding time, the model must decode a document starting with the first paragraph and use its own predictions to compute h^{i-1} and update the memory. Additionally, at decoding time, we assume a five paragraph structure (introduction, three body paragraphs, and conclusion) as a pre-set discourse structure to decode from.

3.6 Experiments

We present experiments comparing PLOTMACHINES with competitive baselines and ablations. In order to investigate as fully as possible, we first provide automatic metrics targeting different aspects of performance (i.e. coverage, diversity). We also use quantitative n-gram analysis to study the usage of outlines. However,

[6]We experimented with a few other variants of implementing multiple attention mechanisms within the transformer blocks, but found this to be empirically effective.

recognizing the limitations of these automatic evaluations (discussed in Chapter 1), we also present human judgments, which are our primary means of analysis. Lastly we discuss some qualitative analysis from output.

3.6.1 Experimental Set-up

Baselines: We compare with two story generation models that have been used in related conditional story generation tasks. First, we train a Fusion model, from the original WritingPrompts dataset paper (Fan et al., 2018), using delimited outlines as a single input in place of a prompt. We also compare with the static storyline-to-story variant of Plan-and-Write (P&W-Static) from Yao et al. (2019). This LSTM-based model was originally used for a task in generating five-line stories from keyword-based storylines. We train the model instead using delimited plot outlines as input.

Additionally, given the recent successes in text generation using large pre-trained LM's, we compare with these models, as well. We finetune the large-scale GROVER (Zellers et al., 2019) (equivalent to GPT-2 medium, 345M param), which is a transformer-based pre-trained language model that has been trained for controllable text generation. We hypothesize that, since this model is trained in a controllable setting, it can also learn to write stories based off of keyword lists. To finetune GROVER, we give the outline as a delimited form of metadata with a new outline token. Because this model is significantly larger than PLOTMACHINES (160M param), we also investigate a 460M parameter variation of PLOTMACHINES that is built on top of GPT-2 medium (Radford et al., 2019) instead of GPT for fuller comparison.

Unlike our models, we train these baselines with the traditional generation framework - i.e. they model and generate an entire document conditioned on outlines without generating each paragraph recurrently.

Ablated PLOTMACHINES Models: We also show results on ablated versions of our model. First, we use the base GPT and GPT2 models, that are fine-tuned similarly to our model but using only outline inputs (without memory, preceding context, or discourse representations). Second, we investigate the effects of using the preceding context representation but still excluding memory and discourse tokens (**PM-NOMEM-NODISC**). Lastly, we use **PM-NOMEM**, a model variant that excludes the memory but still uses outline, discourse, and preceding context representations as input.

Details: We use the HuggingFace implementations of GPT and GPT-2. We finetune all models using

	Wikiplots			WritingPrompts			New York Times		
Model	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
P&W-Static (Yao et al., 2019)	17.0	3.3	13.6	19.2	3.6	14.4	19.3	4.6	15.6
Fusion (Fan et al., 2018)	22.7	6.0	17.4	14.3	1.7	9.6	**23.2**	**7.2**	**18.1**
GROVER (Zellers et al., 2019)	19.6	5.9	12.5	23.7	5.3	17.2	20.0	5.8	14.2
PLOTMACHINES (GPT)	20.2	5.3	16.0	30.5	5.3	25.4	21.2	5.0	15.5
– base (GPT) (Radford et al., 2018)	13.2	2.0	7.9	22.1	2.7	14.3	13.9	1.6	8.3
PLOTMACHINES (GPT-2)	**22.8**	**6.5**	**17.5**	**31.1**	**6.7**	**26.1**	22.1	6.4	16.5
– PM-NOMEM (GPT-2)	20.5	4.9	15.5	26.6	3.7	23.5	20.0	5.4	14.4
– PM-NOMEM-NODISC (GPT-2)	19.3	1.7	13.9	26.8	4.5	23.2	18.4	3.4	14.2
– base (GPT-2) (Radford et al., 2019)	18.5	3.9	13.3	26.5	4.6	20.5	19.2	4.7	13.6

Table 3.2: ROUGE Results on Wiki, WritingPrompts and NYTimes Datasets. The top block represents the baseline models on story/article generation, while the bottom blocks include ablations of our PLOTMACHINES models.

ADAM. For generating with our models, we use nucleus sampling with repetition penalties (Holtzman et al., 2020; Keskar et al., 2019) using $p = 90$ and $\theta = 1.5$ for GPT and $p = 70$ and $\theta = 1.4$ for GPT-2 (based on a hyperparameter sweep with validation data). We use a minimum sequence length of 100 bpe tokens per paragraph and a maximum sequence length of 400, 922 bpe per paragraph for GPT and GPT-2, respectively. We set n, the maximum number of outline tokens and memory dimensions to 100. We used the out-of-the-box settings for Grover/Fusion/Plan-and-write from their respective papers and codebases.

3.6.2 Automatic Metrics

In this section, we evaluate performance using different automatic metrics. We compute ROUGE scores (Lin, 2004) and self-BLEU (Zhu et al., 2018) to follow guidelines from previous work (Shen et al., 2019; Zhu et al., 2018) showing that a low self-BLEU score together with a large ROUGE score can demonstrate a model's ability to generate realistic-looking as well as diverse generations. While these automatic metrics have limitations and are not enough on their own to evaluate modeling effectiveness, we use them as an initial investigation before delving into human-based evaluations.

Coverage We compute ROUGE (Lin, 2004) scores with respect to the gold document (Table 3.2). Results show that the full PLOTMACHINES achieves comparable or higher ROUGE scores on all three datasets. Both PLOTMACHINES variants (using GPT or GPT-2 as a base) achieve significant improvement over GROVER (even though GROVER includes more layers and parameters than the model using GPT).

Ablations In the bottom block of Table 3.2, we compare performance of ablated versions of PLOTMA-

	Wikiplots					Writing Prompts					NY Times				
Model	AvgL	B-2	B-3	B-4	B-5	AvgL	B-2	B-3	B-4	B-5	AvgL	B-2	B-3	B-4	B-5
Gold Test	330	.74	.50	.29	.15	661	.82	.61	.40	.25	315	.73	.50	.32	.21
Fusion	191	.84	.71	.58	.48	197	.93	.85	.75	.65	171	.89	.80	.70	.60
GPT	909	.77	.47	.25	.11	799	.73	.40	.19	.08	739	.68	.36	.27	.08
GPT-2	910	.60	.26	.10	.03	799	.74	.41	.19	.08	756	.69	.36	.17	.08
GROVER	835	.72	.49	.48	.37	997	.88	.72	.52	.34	719	.79	.57	.38	.25
PLOTMACHINES (GPT)	682	.77	.58	.40	.27	850	.89	.81	.72	.63	537	.85	.69	.53	.40
PLOTMACHINES (GPT-2)	553	.56	.19	.07	.02	799	.83	.56	.30	.14	455	.79	.57	.37	.23

Table 3.3: ***Automatic Diversity***: Average Length of the generated test documents (**AvgL**) and Self-BLEU n-gram (**B-n**) scores on 1000 generated story samples from the test datasets of Wikiplots/WritingPrompts/NYTimes. We also include the average length and self-BLEU scores of the gold test data. A lower self-BLEU score together with a large ROUGE (see Table 3.2) score can justify the effectiveness of a model.

CHINES. First, we compare base GPT-2 with PM-NOMEM-NODISC, which differs by including preceding context representations. We observe that PM-NOMEM-NODISC performs slightly better than GPT-2, emphasizing the importance of including context from the previous paragraph. Second, we investigate the impact of discourse structure representations. We compare PM-NOMEM-NODISC, which omits the discourse token, with PM-NOMEM, which uses the discourse token. As shown in Table 3.2, PM-NOMEM generally has higher ROUGE scores than PM-NOMEM-NODISC, indicating that the discourse representation is beneficial to the model. Lastly, we compare PM-NOMEM with the full PLOTMACHINES to determine the effects of having a memory component. The full model with memory has large improvements in ROUGE scores over PM-NOMEM, underscoring the importance of the plot state tracking in this task.

Diversity We evaluate the diversity of generated paragraphs from our models using Self-BLEU scores (Zhu et al., 2018). In Table 3.3, we report the self-BLEU scores (Zhu et al., 2018) along with the average length of each generated story. Using all the generated documents from a model, we take one generated document as hypothesis and the others as reference, and calculate BLEU score for every generated document, and define the average BLEU score to be the self-BLEU of the model. A lower self-BLEU score together with a large ROUGE score can justify the effectiveness of a model (Shen et al., 2019; Zhu et al., 2018), *i.e.*, being able to generate realistic-looking as well as diverse generations across the selected test documents. For example, the Fusion model achieved relatively high ROUGE scores, but it has generally poor diversity scores (much higher self-BLEU than all the other models in Table 3.3). On closer examination, it seems that this model may be achieving high performance in the overlap-based ROUGE scores by producing text that is more repetitive and generic. In contrast, PLOTMACHINES generally achieves good performance on both

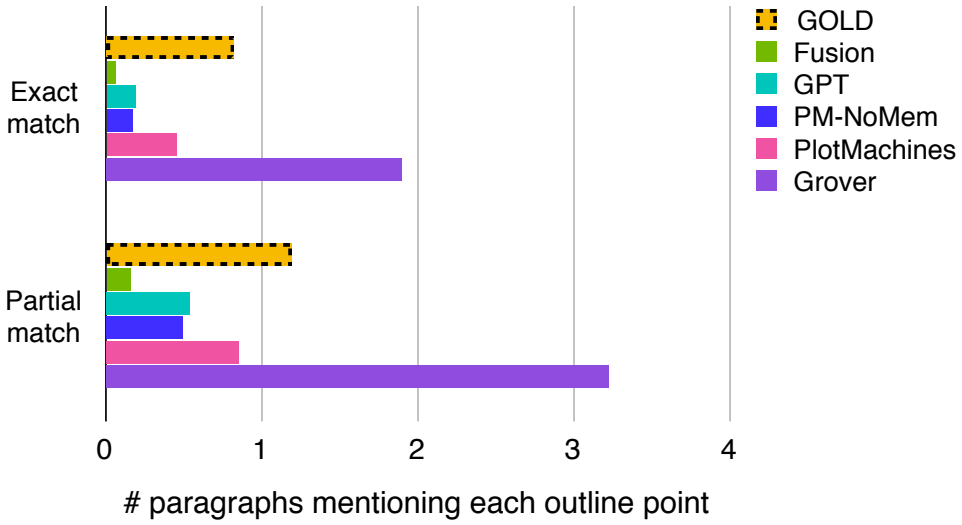


Figure 3.3: Number of paragraphs that mention each outline point. In contrast to base GPT and Fusion, the PLOTMACHINES model with memory has better coverage of outline points, more similarly to the gold document. GROVER, meanwhile, tends to repeat outline points in more paragraphs (even significantly more than the gold reference).

ROUGE and diversity scores, with self-BLEU scores that are lower than most other models. Notably, they often have more similar self-BLEU scores to the actual gold stories, indicating that the language diversity is more similar to what humans write.

3.6.3 N-gram Based Outline Usage Analysis

We perform additional quantitative analysis to see how many times outline points are included across an entire document. For fifty stories in the Wikiplots validation set, we compute how many paragraphs mention each outline point using exact matching or partial matching (in which $> 20\%$ of the n-grams in the outline point also appear in the paragraph). We report the results in Figure 3.3.

Notably, GROVER tends to repeat outline points several times (about twice as much as the actual gold paragraphs). Similar observations have been made about large pre-trained language models in See et al. (2019) that the models could be used to follow story prompts very closely but often copied too much compared to human writing.

In contrast, the Fusion model tends to leave out portions of the outline. This may reflect the way Fusion was originally designed – for use with a task using more abstract prompts as input. The GPT and PM-

NOMEM models, while more inclusive than Fusion, are also likely to exclude outline points. Our full PM model is generally more inclusive and more similar to the gold reference than the other models. The gold story mentions each outline point in around one paragraph on average, indicating that there is an ideal balance between the more conservative coverage achieved by our model and the over-repetitive coverage of GROVER.

3.6.4 Human Evaluations

Due to limitations of automatic metrics (Liu et al., 2016), we conduct two human studies to further explore how generated stories compare along three dimensions: outline utilization, narrative flow, and ordering. We ask three people[7] to rate each generation or pair of generations.

In the first experiment, we run a small study where humans rate pairs of stories for 20 randomly sampled stories per pair of models. We ask them detailed questions about how the two stories compare across the three dimensions. To scale up to a larger study, we run a second experiment asking humans to rate much shorter story excerpts for 100 randomly sampled test set stories per model. In these experiments, we use stories and excerpts from the Wikiplots test set.

Full Story Ratings

In this task, we give human raters a pair of stories generated from the same outlines and ask them to choose which one is better in different categories related to outline utilization, narrative flow, and ordering. In Figure 3.4, we show how often PLOTMACHINES was selected over the other two models (values above 50% indicate that PLOTMACHINES was preferred). PLOTMACHINES was selected over base GPT in all of the categories, demonstrating that the memory and discourse features are vitally

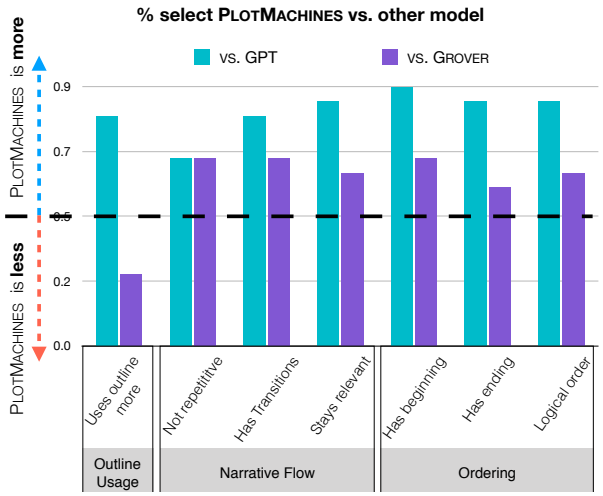


Figure 3.4: Head-to-head comparison of PLOTMACHINES vs. two other models for full stories.

[7] using Amazon Mechanical Turk.

important to improving the base model. Although humans rated GROVER as using the outline more, this may be a side-effect of our findings in Sec. 3.6.3 that GROVER tends to over-repeat key points (twice as much as gold). PLOTMACHINES is ranked higher in all of the questions about narrative flow and ordering.

Excerpt Ratings

Outline Usage We give raters two paragraphs each generated by different models and ask them to select which is utilizing the outline better. We perform two trials, one with random paragraphs from each story and one with the paragraph from each story that has the most n-gram overlap with the outline (i.e. the closest).

Outline Utilization		
Model A	Model B	% Prefer Model A
Random Paragraph		
PLOTMACHINES	GPT	**72**%
PLOTMACHINES	GROVER	49%
Closest Paragraph		
PLOTMACHINES	GPT	**83**%
PLOTMACHINES	GROVER	**54**%

Table 3.4: Humans judge which of two paragraphs better utilize the outlines (when shown either random paragraphs or the paragraphs most similar to the outline).

Results in Table 3.4 show that humans may select PLOTMACHINES and GROVER comparably in both set-ups but may have a slight preference towards PLOTMACHINES when looking at the "closest" paragraph from both models. In both set-ups, humans say that paragraphs from PLOTMACHINES use the outline better than the base GPT model.

Narrative Flow In this task, we give raters a generated paragraph (with the previous paragraph as context). They are asked to rate on a Likert from 1-5 scale how much the paragraph: (1) repeats content from the previous paragraph, (2) transitions naturally from the previous paragraph, and (3) stays relevant (on-topic) throughout the paragraph.

Model	Narrative Flow			Order
	Rep(↓)	Tran(↑)	Rel(↑)	Acc(↑)
GPT	**1.39**	1.89	2.06	42
GROVER	1.78	3.00	3.29	**62**
PM	1.64	**3.02**	**3.39**	59

Table 3.5: Human evaluations of paragraph excerpts from GPT, GROVER and PLOTMACHINES (PM) outputs.

In the left side of Table 3.5, we show the average ratings of each model. Base GPT is the least repetitive between paragraph. However, this may be caused

by generating unrelated content from one paragraph to the next, as corroborated by the low subscores of the transitions and relevance. The model rated most repetitive is GROVER, mirroring observations about the repetitive nature of that model in how it uses key points, discussed in Sec. 3.6.3. The transitions between two paragraphs are rated similarly highly between GROVER and PLOTMACHINES models. However, PLOTMACHINES tends to achieve highest relevancy within paragraphs.

Ordering We give raters a pair of consecutive generated paragraphs in a random order and ask them order the paragraphs. We compute the accuracy of the majority vote compared to the actual order in the right side of Table 3.5. Human accuracy for PLOTMACHINES models are approaching 60% accuracy and are significantly better than the base GPT. GROVER is easiest for humans to re-order (62%). This result differs slightly from the full story analysis where the humans preferred PLOTMACHINES over GROVER in all of the ordering-based questions. One possible explanation is that GROVER, which decodes the entire document token-by-token (rather than decoding each paragraph recurrently) may be better at resolving problems like coreference between paragraphs. This makes it easier for humans to re-order short excerpts even though they generally prefer the overall narrative order of PLOTMACHINES due to it having better beginnings, endings, etc. (as indicated in our full story human study).

3.6.5 Qualitative Analysis

To verify the learned discourse structure, we show examples of introduction and conclusion paragraphs generated by PLOTMACHINES (Table 3.6). For example, the model often starts stories by setting the scene (e.g. "In the early 1950s, a nuclear weapons testing continues ....") and often ends with a definitive closing action (e.g. "... the film ends with humperdinck and buttercup riding off into the sunset."). This indicates that the model is learning the appropriate discourse associated with different types of paragraphs.

3.7 Summary

In this chapter, we discuss plot dynamics in narratives. We present outline-conditioned story generation, a new task for generating stories from outlines containing key plot elements. We facilitate training for this task by constructing three datasets with outlines for multi-paragraph stories. In order to better incorporate plot

Paragraph type	Paragraph
intro	in the early 1950s, a nuclear weapons testing continues at an underwater hydrogen bomb test site. scientists are concerned that it may be too dangerous to detonate without being detected by radar and radiation detectors. government sends paleontologist kyohei yamane (kim kap) to investigate. he is killed when his boat explodes while on shore patrol. as evidence describes damage consistent with sabotage of oil rigs, they conclude there must have been more than one way inside the facility. meanwhile, military research has discovered a deep underwater natural habitat alongside others where water can not be mined for life - saving purposes.
intro	the novel is set in a post - apocalyptic future where earth almost uninhabitable, with only one habitable planet for habitation and an intersolar system police force (rf) to maintain order. the story begins when " cowboy bebop ", who has been living on his homeworld of nepal since he was 12 years old, returns from space after being stranded by a comet that destroyed most of the interstellar civilization. he finds himself at home as well as friends among other characters.
conclusion	humperdinck arranges for buttercup to get married to a powerful don juan carlos, who is rumored to be able to control the entire province. humperdinck secretly orders rugen to kidnap buttercup and bring her to him. rugen succeeds in kidnapping buttercup, but humperdinck kidnaps her anyway. buttercup manages to free herself and flee with humperdinck, but is captured by manuela, who accuses humperdinck of trying to keep her prisoner. humperdinck swears revenge on manuela and his henchmen, and rescues buttercup just in time. the pair head north to santa fe, where humperdinck uses his magic powers to heal buttercup 's wounds. the couple settle in a small cabin owned by mrs mccluskey, who introduces buttercup to mr smith, a blacksmith. humperdinck 's plan backfires when mr smith is attacked by apache indians, and humperdinck saves him. the film ends with humperdinck and buttercup riding off into the sunset .
conclusion	stevens and angel eyes sneak into the church hall and steal a bible. stevens opens the book and reads passages from psalms 118 to 350 bc. stevens closes the book and hands it to angel eyes. angel eyes then places stevens ' hand atop the cross and prepares to strike. stevens grabs hold of angel eyes and begs him to reconsider. stevens pleads with angel eyes to listen to reason. angel eyes makes stevens tell him why he left the confederacy. stevens tells him that he was betrayed by his mother and sister and that he needs redemption. stevens then lies and tells angel eyes that he ca n't forgive him. stevens then walks away. angel eyes watches him disappear into the night .

Table 3.6: Example introduction and conclusion paragraph generations from PLOTMACHINES using Wikiplots validation set.

elements, we create PLOTMACHINES which generates paragraphs using a high-level discourse structure and a dynamic plot memory that keeps track of the outline and story. Quantitative analysis shows that PLOTMACHINES is effective in composing tighter narratives based on outlines compared to competitive baselines.

Chapter 4

Investigating Writer's Intent with Connotation Frames

People often convey implicit messages about social relationships through word choice and phrasing. Being able to understand the underlying connotation of a piece of text is an integral part of understanding the implicit intent of the writer. In this chapter, we formalize a set of connotation frames based on predicate word choice. More concretely, the connotation frame of a particular verb predicate defines various implied relationships towards its semantic arguments. We annotate a set of a thousand verb predicates for nine types of relations conferred to the agent and theme of the verb. We also demonstrate that using connotation frames could have applications for tasks requiring nuanced understanding of text. As an example, we show that large-scale connotative analysis of news text can be used to automatically reveal implied partisan stance towards specific political issues. This chapter includes work originally published in Rashkin et al. (2016).

4.1 Introduction

People commonly express their opinions through subtle and nuanced language (Thomas et al., 2006; Somasundaran and Wiebe, 2010). Often, through seemingly objective statements, the writer can influence the readers' judgments toward an event and its participants. Even by choosing a particular predicate, the writer can indicate rich connotative information about the entities that are interacting. For example, through a

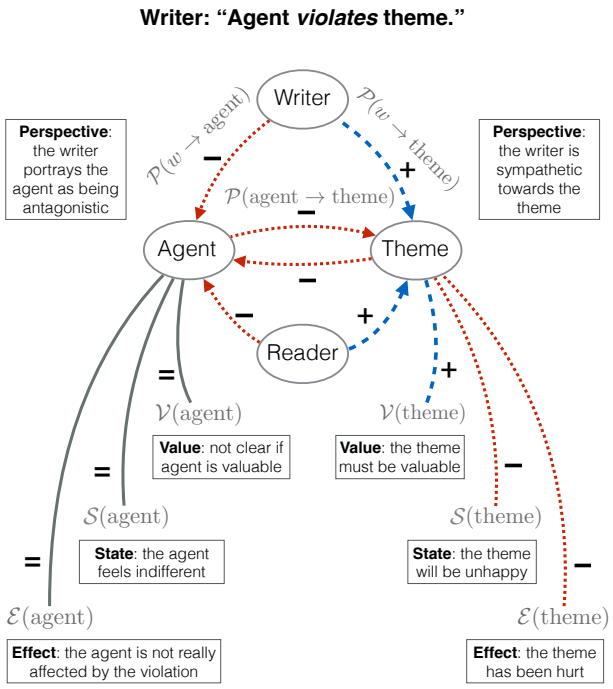


Figure 4.1: An example connotation frame of "violate" as a set of typed relations: perspective $\mathcal{P}(x \rightarrow y)$, effect $\mathcal{E}(x)$, value $\mathcal{V}(x)$, and mental state $\mathcal{S}(x)$.

simple statement such as "x violated y", the writer can convey:

(1) writer's perspective: the writer is portraying x as an "antagonist" and y as a "victim", eliciting negative perspective from readers toward x (i.e., blaming x) and positive perspective toward y (supportive or sympathetic to y).

(2) entities' perspective: y most likely feels negatively toward x as a result of being violated.

(3) effect: something bad happened to y.

(4) value: y is something valuable, since it does not make sense to violate something worthless. In other words, the writer is presupposing y's positive value as a fact.

(5) mental state: y is most likely unhappy about the outcome.[1]

Even though the writer might not explicitly state any of the interpretations [1-5] above, the readers will be able interpret these intentions as a part of their comprehension. In this chapter, we present an empirical study of how to represent and induce the connotative interpretations that can be drawn from a verb predicate, as illustrated above.

[1]To be more precise, y is most likely in a negative mental state assuming it is an entity that can have a mental state.

We introduce *connotation frames* as a representation framework to organize the rich dimensions of the implied sentiment and presupposed facts. Figure 4.1 shows an example of a connotation frame for the predicate *violate*. We define four different typed relations: $\mathcal{P}(x \rightarrow y)$ for perspective of x towards y, $\mathcal{E}(x)$ for effect on x, $\mathcal{V}(x)$ for value of x, and $\mathcal{S}(x)$ for mental state of x. These relationships can all be either positive (+), neutral (=), or negative (-).

The work outlined in this chapter is novel in its investigation of frames as a representation formalism for connotative meanings. This contrasts with previous computational studies and resource development for frame semantics, where the primary focus was almost exclusively on denotational meanings of language (Baker et al., 1998; Palmer et al., 2005). However, our formalism draws inspiration from these earlier works in that we investigate the connection between a word and relevant associations from world knowledge (Fillmore, 1976), which is essential for the readers to interpret many layers of the implied sentiment and presupposed value judgments.

We also build upon the extensive amount of literature in sentiment analysis (Pang and Lee, 2008; Liu and Zhang, 2012), especially the recent efforts in implied sentiment analysis (Feng et al., 2013; Greene and Resnik, 2009), entity-entity sentiment inference (Wiebe and Deng, 2014), opinion role induction (Wiegand and Ruppenhofer, 2015), and effect analysis (Choi and Wiebe, 2014). However, we are the first to organize aspects of the connotative information into coherent frames.

To facilitate further study in connotation frames, we introduce a new formalism, model, and annotated dataset for studying the connotation frames from large-scale natural language data and statistics. We also discuss data-driven insights into the dynamics among different typed relations within each frame. Finally, we include an analysis of stance described in news text to show one potential use of connotation frames.

4.2 Related Work

Most prior work on sentiment lexicons focused on the overall polarity of words without taking into account their semantic arguments (Wilson et al., 2005; Baccianella et al., 2010; Wiebe et al., 2005; Velikovich et al., 2010; Kaji and Kitsuregawa, 2007; Kamps et al., 2004; Takamura et al., 2005; Adreevskaia and Bergler, 2006). Several recent studies explore more specific and nuanced aspects of sentiment such as connotation (Feng et al., 2013), good and bad effects (Choi and Wiebe, 2014), and evoked sentiment (Mohammad

and Turney, 2010). Another related line of work is the study of near-synonyms that also investigates the intention of a word choice with distinction between similar words (Edmonds and Hirst, 2002). Drawing inspirations from them, we present connotation frames as a unifying representation framework to encode the rich dimensions of implied sentiment, presupposed value judgements, and effect evaluation, and propose a factor graph formulation that captures the interplay among different types of connotation relations.

Goyal et al. (2010a,b) investigated how characters (protagonists, villains, victims) in children's stories are affected by certain predicates, which is related to the effect relations encoded in connotation frames. While Klenner et al. (2014) similarly investigated the relation between the polarity of the verbs and arguments, our work introduces new perspective types and proposes a unified representation and inference model. Wiegand and Ruppenhofer (2015) also looked at perspective-based relationships induced by verb predicates with a focus on opinion roles. Building on this concept, our framework also incorporates information about the perspectives' polarities as well as information about other typed relations. There have been growing interests for modeling framing (Greene and Resnik, 2009; Hasan and Ng, 2013), biased language (Recasens et al., 2013) and ideology detection (Yano et al., 2010). All these tasks are relatively less studied, and we hope our connotation frame lexicon will be useful for them.

Sentiment inference rules have been explored by Wiebe and Deng (2014) and Deng and Wiebe (2014). The focus of their work was on general inference rules that are not predicate-specific. In contrast, our work focuses on the notion that connotative polarities can be determined directly from the predicate, rather than partial knowledge of the arguments or the context in which it is being used. In brief, we make a novel conceptual connection between inferred sentiments and frame semantics, organized as connotation frames, and present a unified model that integrates different aspects of the connotation frames.

Finally, in a broader sense, what we study as connotation frames draws a connection to schema and script theory (Schank and Abelson, 1975). Unlike prior work that focused on directly observable actions (Chambers and Jurafsky, 2009; Frermann et al., 2014; Bethard et al., 2008), we focus on implied sentiments that are framed by predicate verbs.

Verb	Subset of Typed Relations		Example Sentences	L/R
suffer	$\mathcal{P}(w \rightarrow \text{agent}) = +$ $\mathcal{P}(w \rightarrow \text{theme}) = -$ $\mathcal{P}(\text{agent} \rightarrow \text{theme}) = -$	$\mathcal{E}(\text{agent}) = -$ $\mathcal{V}(\text{agent}) = +$ $\mathcal{S}(\text{agent}) = -$	The story begins in Illinois in 1987, when *a 17-year-old girl* **suffered** *a botched abortion.*	R
guard	$\mathcal{P}(w \rightarrow \text{agent}) = +$ $\mathcal{P}(w \rightarrow \text{theme}) = +$ $\mathcal{P}(\text{agent} \rightarrow \text{theme}) = +$	$\mathcal{E}(\text{theme}) = +$ $\mathcal{V}(\text{theme}) = +$ $\mathcal{S}(\text{theme}) = +$	In August, *marshals* **guarded** *25 clinics* in 18 cities.	L
uphold	$\mathcal{P}(w \rightarrow \text{theme}) = +$ $\mathcal{P}(\text{agent} \rightarrow \text{theme}) = +$	$\mathcal{E}(\text{theme}) = +$ $\mathcal{V}(\text{theme}) = +$	A hearing is scheduled to make a decision on whether to **uphold** *the clinic's suspension.*	R

Table 4.1: Example typed relations (perspective $\mathcal{P}(x \rightarrow y)$, effect $\mathcal{E}(x)$, value $\mathcal{V}(x)$, and mental state $\mathcal{S}(x)$), where w denotes the writer. Not all typed relations are shown due to space constraints. The example sentences demonstrate the usage of the predicates in left [L] or right [R] leaning news sources.

Verb	x's role	$\mathcal{P}(\mathbf{w} \rightarrow \cdot)$	Left-leaning Sources	Right-leaning Sources
accuse	agent	-	Putin, Progressives, Limbaugh, Gingrich	activist, U.S., protestor, Chavez
	theme	+	official, rival, administration, leader	Romney, Iran, Gingrich, regime
attack	agent	-	McCain, Trump, Limbaugh	**Obama**, campaign, Biden, Israel
	theme	+	Gingrich, **Obama**, policy	citizen, Zimmerman
criticize	agent	-	Ugandans, rival, Romney, Tyson	Britain, passage, **Obama**, Maddow
	theme	+	**Obama**, Allen, Cameron, Congress	Pelosi, Romey, GOP, Republicans

Table 4.2: Media Bias in Connotation Frames: Obama, for example, is portrayed as someone who *attacks* or *criticizes* others by the right-leaning sources, whereas the left-leaning sources portray Obama as the victim of harsh acts like "attack" and "criticize".

4.3 Connotation Frame

Given a predicate v, we define a connotation frame $\mathcal{F}(v)$ as a collection of typed relations and their polarity assignments: (i) **perspective** $\mathcal{P}^v(x_i \rightarrow x_j)$: whether the predicate v implies directed sentiment from the entity x_i to the entity x_j, (ii) **value** $\mathcal{V}^v(x_i)$: whether x_i is presupposed to be valuable by the predicate v, (iii) **effect** $\mathcal{E}^v(x_i)$: whether the event denoted by the predicate v is good or bad for the entity x_i, and (iv) **mental state** $\mathcal{S}^v(x_i)$: the likely mental state of the entity x_i as a result of the event. We assume that each typed relation can have one of the three connotative polarities $\in \{+, -, =\}$, i.e., positive, negative, or neutral. Our goal is to focus on the general connotation of the predicate considered out of context. Future work may look into contextual interpretation of connotation as well.

Table 4.1 shows examples of connotation frame relations for the verbs *suffer*, *guard*, and *uphold*, along with example sentences. For instance, for the verb *suffer*, the writer is likely to have a positive perspective towards the agent (e.g., being supportive or sympathetic toward the "17-year-old girl" in the example shown

on the right) and a negative perspective towards the theme (e.g., being negative towards 'botched abortion").

4.3.1 Data-driven Motivation

Since the meaning of language is ultimately contextual, the exact connotation will vary depending on the context of each utterance. Nonetheless, there still are common shifts or biases in the connotative polarities, as we see in data-driven analyses.

First, we looked at words from the Subjectivity Lexicon (Wilson et al., 2005) that are used in the argument positions of a small selection of predicates in Google Syntactic N-grams (Goldberg and Orwant, 2013). For this analysis, we assumed that the agent is the word in the subject position while the theme is the word in the object position. We found 64% of the words in the agent role of *suffer* are positive, and 94% of the words in the theme role are negative, which is consistent with the polarities of the writer's perspective towards these arguments, as shown in Table 4.1. For *guard*, 57% of the agents and 76% of the themes are positive, and in the case of *uphold*, 56% of the agents and 72% of the themes are positive.

We also investigated how media bias can potentially be analyzed through connotation frames. From the Stream Corpus 2014 dataset (KBA, 2014), we selected all articles from news outlets with known political biases,[2] and compared how they use certain verbs such as "accuse", "attack", and "criticize" differently. We see how the usages compares in light of the $\mathcal{P}(w \rightarrow agent)$ and $\mathcal{P}(w \rightarrow theme)$ relations of the connotation frames. Table 4.2 shows interesting contrasts. Obama, for example, is frequently portrayed as someone who *attacks* or *criticizes* others according to the right-leaning sources, whereas the left-leaning sources portray Obama as the victim of harsh acts like "attack" or "criticize".[3] Furthermore, by knowing the perspective relationships $\mathcal{P}(w \rightarrow x_i)$ associated with a predicate, we can make predictions about how the left-leaning and right-leaning sources feel about specific people or issues. For example, because left-leaning sources frequently use McCain, Trump, and Limbaugh in the agent position of "attack", we might predict that these sources are less likely to be sympathetic towards these entities.

[2]The articles come from 30 news sources indicated by others as exhibiting liberal or conservative leanings (Mitchell et al., 2014; Center for Media and Democracy, 2013, 2012; HWC Library, 2011)

[3]This may mean that even when describing similar events, right and left-learning sources might choose slightly different wordings to support their portrayal of Obama.

Perspective Triad: If argument x_i is positive towards x_j, and x_j is positive towards x_k, then we expect x_i is also positive towards x_k. Similar dynamics hold for the negative case. $$\mathcal{P}_{w \to x_i} = \neg \left(\mathcal{P}_{w \to x_j} \oplus \mathcal{P}_{x_i \to x_j} \right)$$
Perspective – Effect: If a predicate has a positive effect on one argument, then we expect that the interaction between the arguments was positive. Similar dynamics hold for the negative case. $$\mathcal{E}_{x_i} = \mathcal{P}_{x_j \to x_i}$$
Perspective – Value: If argument x_i is presupposed as valuable, then we expect that the writer also views x_i positively. Similar dynamics hold for the negative case. $$\mathcal{V}_{x_i} = \mathcal{P}_{w \to x_i}$$
Effect – Mental State: If the predicate has a positive effect on an argument x_i, then we expect that x_i will gain a positive mental state. Similar dynamics hold for the negative case. $$\mathcal{S}_{x_i} = \mathcal{E}_{x_i}$$

Table 4.3: Potential Dynamics among Typed Relations: we propose models that parameterize these dynamics using log-linear models (frame-level model in §3).

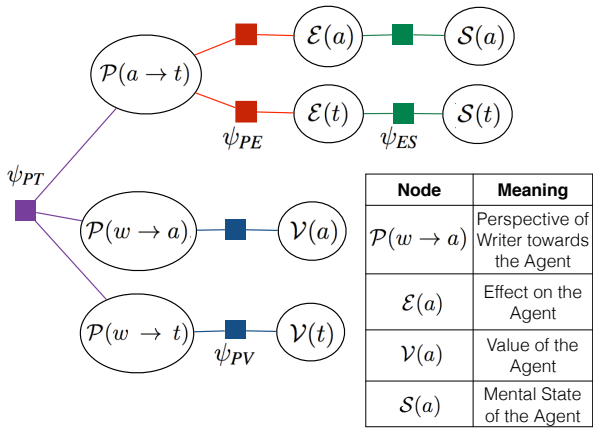
4.3.2 Dynamics Between Typed Relations

Given a predicate, the polarity assignments of typed relations are interdependent. For example, if the writer feels positively towards the agent but negatively towards the theme, then it is likely that the agent and the theme do not feel positively towards each other. This insight is related to types of implicature investigated in Wiebe and Deng (2014), but differs in that the polarities are predicate-specific and do not rely on knowledge of prior sentiment towards the arguments, themselves. This and other possible interdependencies are summarized in Table 4.3. These interdependencies serve as general guidelines of what properties we expect to depend on one another, especially in the case where the polarities are non-neutral. We will promote these internal consistencies in our factor graph model (§4.4) as soft constraints.

There also exist other interdependencies that we use to simplify our task. First, the directed sentiments between the agent and the theme are likely to be reciprocal, or at least do not directly conflict with $+$ and $-$ simultaneously. This intuition follows from a notion of balance derived by social theory (Heider, 1946). Therefore, we assume that $\mathcal{P}(x_i \to x_j) = \mathcal{P}(x_j \to x_i) = \mathcal{P}(x_i \leftrightarrow x_j)$, and we only measure for these binary relationships going in one direction. In addition, we assume the *predicted*[4] perspective from the reader r to an argument, $\mathcal{P}(r \to x)$, is likely to be the same as the *implied* perspective from the writer w to the same argument, $\mathcal{P}(w \to x)$. So, we only try to learn the perspective of the writer.

For simplicity, our work only explores verb predicates and focuses on the polarities involving the agent and the theme roles, which we will refer to as a and t. We assume that these roles are correlated to the

[4]Surely different readers can and will form varying opinions after reading the same text. Here we concern with the most likely perspective of the general audience, as a result of reading the text.



Node	Meaning
$\mathcal{P}(w \rightarrow a)$	Perspective of Writer towards the Agent
$\mathcal{E}(a)$	Effect on the Agent
$\mathcal{V}(a)$	Value of the Agent
$\mathcal{S}(a)$	Mental State of the Agent

Figure 4.2: A factor graph for predicting the polarities of the typed relations that define a connotation frame for a given verb predicate. The factor graph also includes unary factors (ψ_{emb}), which we left out for brevity.

subject and object positions.

4.4 Modeling Connotation Frames

Our task is essentially that of lexicon induction (Akkaya et al., 2009; Feng et al., 2013) in that we want to predict the connotation frames of previously unseen verbs. For each verb predicate, we infer a connotation frame composed of 9 relationship aspects that represent: *perspective* $\{\mathcal{P}(w \rightarrow t), \mathcal{P}(w \rightarrow a), \mathcal{P}(a \rightarrow t)\}$, *effect* $\{\mathcal{E}(t), \mathcal{E}(a)\}$, *value* $\{\mathcal{V}(t), \mathcal{V}(a)\}$, and *mental state* $\{\mathcal{S}(t), \mathcal{S}(a)\}$ polarities, where w, a, t denote the writer, the agent, and the theme, respectively.

We propose two models: an aspect-level model that makes the prediction for each typed relation independently based on the distributional representation of the context in which the predicate commonly appears (§4.4.1), and a frame-level model that makes the prediction over the connotation frame collectively in consideration of the dynamics between typed relations (§4.4.2).

4.4.1 Aspect-Level

Our aspect-level model predicts labels for each of these typed relations separately. As input, we use the 300-dimensional dependency-based word embeddings from Levy and Goldberg (2014). For each aspect, there is a separate MaxEnt (maximum entropy) classifier used to predict the label of that aspect on a given word-embedding, which is treated as a 300 dimensional input vector to the classifier. The MaxEnt classifiers

learn their weights using LBFGS on the training data examples with re-weighting of samples to maximize for the best average F1 score.

4.4.2 Frame-Level

Next we present a factor graph model (Figure 4.2) of the connotation frames that parameterizes the dynamics between typed relations. Specifically, for each verb predicate, the factor graph contains 9 nodes representing the different aspects of the connotation frame involving the writer (w), the agent (a), and the theme (t). All these variables take polarity values from the set $\{-, =, +\}$.

We define $\mathbf{Y}_i := \{\mathcal{P}_{wt}, \mathcal{P}_{wa}, \mathcal{P}_{at}, \mathcal{E}_t, \mathcal{E}_a, \mathcal{V}_t, \mathcal{V}_a, \mathcal{S}_t, \mathcal{S}_a\}$ as the set of relational aspects for the i^{th} verb predicate. The factor graph for $\mathbf{Y}_i$, is illustrated in Figure 4.2, and we describe the factor potentials, ψ, in detail in the rest of this section. The probability of an assignment of polarities to the nodes in $\mathbf{Y}_i$ is:

$$\begin{aligned} P(\mathbf{Y}_i) \;\propto\; & \psi_{\text{PV}}(\mathcal{P}_{wa}, \mathcal{V}_a)\; \psi_{\text{PV}}(\mathcal{P}_{wt}, \mathcal{V}_t) \\ & \psi_{\text{PE}}(\mathcal{P}_{at}, \mathcal{E}_a)\; \psi_{\text{PE}}(\mathcal{P}_{at}, \mathcal{E}_t) \\ & \psi_{\text{ES}}(\mathcal{E}_a, \mathcal{S}_a)\; \psi_{\text{ES}}(\mathcal{E}_t, \mathcal{S}_t) \\ & \psi_{\text{PT}}(\mathcal{P}_{wt}, \mathcal{P}_{wa}, \mathcal{P}_{at}) \prod_{y \in \mathbf{Y}_i} \psi_{emb}(y) \end{aligned}$$

Embedding Factors We include unary factors on all nodes to represent the results of the aspect-level classifier. Incorporating this knowledge as factors, as opposed to *fixing* the variables as observed, affords us the flexibility of representing noise in the labels as soft evidence. The potential function ψ_{emb} is a log-linear function of a feature vector f, which is a one-hot feature vector representing the polarity of a node ($+$,$-$,or $=$) according to the aspect-level prediction. For example, with the node representing the value of the theme ($\mathcal{V}_t$):

$$\psi_{emb}(\mathcal{V}_t) = e^{\theta_{\mathcal{V}_t} \cdot f(\mathcal{V}_t)}$$

The potential ψ_{emb} is defined similarly for the remaining eight nodes.

Weights θ are learned in a piecewise likelihood manner (Sutton and McCallum, 2009) for each factor

independently using stochastic gradient descent (SGD) over the training data.

Interdependency Factors We include interdependency factors to promote the properties defined by the dynamics between relations (§4.3.2). The potentials for Perspective Triad, Perspective-Value, Perspective-Effect, and Effect-State Relationships (ψ_{PT}, ψ_{PV}, ψ_{PE}, ψ_{ES} respectively) are all defined using log-linear functions of one-hot feature vectors that encode the combination of polarities of the neighboring nodes. Thus the potential ψ_{PT} is:

$$\psi_{PT}(\mathcal{P}_{wt}, \mathcal{P}_{wa}, \mathcal{P}_{at}) = e^{\theta_{PT} \cdot f(\mathcal{P}_{wt}, \mathcal{P}_{wa}, \mathcal{P}_{at})}$$

And we define the potentials for ψ_{PV}, ψ_{PE}, and ψ_{ES} for nodes pertaining to the agent as:

$$\psi_{PV}(\mathcal{P}_{wa}, \mathcal{V}_a) = e^{\theta_{PV,a} \cdot f(\mathcal{P}_{wa}, \mathcal{V}_a)}$$

$$\psi_{PE}(\mathcal{P}_{at}, \mathcal{E}_a) = e^{\theta_{PE,a} \cdot f(\mathcal{P}_{at}, \mathcal{E}_a)}$$

$$\psi_{ES}(\mathcal{E}_a, \mathcal{S}_a) = e^{\theta_{ES,a} \cdot f(\mathcal{E}_a, \mathcal{S}_a)}$$

and we define the potentials for the theme nodes similarly. As with the unary seed factors, weights θ are learned using SGD over training data.

Belief Propagation We use belief propagation to induce the connotation frames of previously unseen verbs. In the belief propagation algorithm, messages are iteratively passed between the nodes to their neighboring factors. Each message μ, containing a scalar for each value $x \in \{-, =, +\}$, is defined from each node v to a neighboring factor f as follows:

$$\mu_{v \to f}(x) \propto \prod_{f^* \in N(v) \setminus a} \mu_{f^* \to v}(x)$$

and from each factor a to a neighboring node v as:

$$\mu_{f \to v} \propto \sum_{x', x'_v = x} \psi(x') \prod_{v^* \in N(f) \setminus v} \mu_{v^* \to f}(x'_{v^*})$$

Our factor graph does not contain any loops, so we are able to perform exact inference by choosing a root node and performing message passing from the leaves to the root and back to the leaves. At the conclusion of message passing, the probability of a specific polarity associated with node v being equal to x is proportional to $\prod_{f \in N(v)} \mu_{f \to v}(x)$.

4.5 Experiments

We first describe crowd-sourced annotations (§4.5.1), then present the empirical results of predicting connotation frames (§4.5.2), and conclude with qualitative analysis of a large corpus (§4.5.3).

4.5.1 Data and Crowdsourcing

In order to understand how humans interpret connotation frames, we designed an Amazon Mechanical Turk (AMT) annotation study. We gathered a set of transitive verbs commonly used in the New York Times corpus (Sandhaus, 2008), selecting the 2400 verbs that are used more than 200 times in the corpus. Of these, AMT workers annotated the 1000 most frequently used verbs.

Annotation Design In a pilot annotation experiment, we found that annotators have difficulty thinking about subtle connotative polarities when shown predicates without any context. Therefore, we designed the AMT task to provide a generic context as follows. We first split each verb predicate into 5 separate tasks that each gave workers a different generic sentence using the verb. To create generic sentences, we used Google Syntactic N-grams (Goldberg and Orwant, 2013) to come up with a frequently seen Subject-Verb-Object tuple which served as a simple three-word sentence with generic arguments.[5] For each of the 5 sentences, we asked 3 annotators to answer questions like "How do you think the agent feels about the event described in this sentence?" In total, each verb has 15 annotations aggregated over 5 different generic sentences containing the verb.

In order to help the annotators, some of the questions also allowed annotators to choose sentiment using additional classes for "positive or neutral" or "negative or neutral" for when they were less confident but still felt like a sentiment might exist. When taking inter-annotator agreement, we count "positive or neutral"

[5]Because Google Syntactic N-grams only provide dependency types and do not provide semantic roles, we approximate the agent and theme as the subject and the object respectively.

as agreeing with either "positive" or "neutral" classes.

Annotator agreement Table 4.4 shows agreements and data statistics. The non-conflicting (NC) agreement only counts opposite polarities as disagreement.[6] From this study, we can see that non-expert annotators are able to see these sort of relationships based on their understanding of how language is used. From the NC agreement, we see that annotators do not frequently choose completely opposite polarities, indicating that even when they disagree, their disagreements are based on the degree of connotations rather than the polarity itself. The average Krippendorff alpha for all of the questions posed to the workers is 0.25, indicating stronger than random agreement. Considering the subtlety of the implicit sentiments that we are asking them to annotate, it is reasonable that some annotators will pick up on more nuances than others. Overall, the percent agreement is encouraging that the connotative relationships are visible to human annotators.

Aspect	% Agreement		Distribution	
	Strict	NC	% +	% -
$\mathcal{P}(w \rightarrow t)$	75.6	95.6	36.6	4.6
$\mathcal{P}(w \rightarrow a)$	76.1	95.5	47.1	7.9
$\mathcal{P}(a \rightarrow t)$	70.4	91.9	45.8	5.0
$\mathcal{E}(t)$	52.3	94.6	50.3	20.24
$\mathcal{E}(a)$	53.5	96.5	45.1	4.7
$\mathcal{V}(t)$	65.2	-	78.64	2.7
$\mathcal{V}(a)$	71.9	-	90.32	1.4
$\mathcal{S}(t)$	79.9	98.0	12.8	14.5
$\mathcal{S}(a)$	70.4	92.5	50.72	8.6

Table 4.4: Label Statistics: % Agreement refers to pairwise inter-annotator agreement. The strict agreement counts agreement over 3 classes ("positive or neutral" was counted as agreeing with either + or neutral), while non-conflicting (NC) agreement also allows agreements between neutral and -/+ (no direct conflicts). Distribution shows the final class distribution of -/+ labels created by averaging annotations.

Aggregating Annotations We aggregated over crowdsourced labels (fifteen annotations per verb) to create a polarity label for each aspect of a verb.[7]

Final distributions of the aggregated labels are included in the right-hand columns of Table 4.4. Notably, the distributions are skewed toward positive and neutral labels. The most skewed connotation frame aspect is the value $\mathcal{V}(x)$ which tends to be positive, especially for the agent argument. This makes some intuitive sense since, as the agent actively causes the predicate event to occur, they most likely have some intrinsic potential to be valuable. An example of a verb where the agent was labelled with negative value is "contaminate".

[6] Annotators were asked yes/no questions related to Value, so this does not have a corresponding NC agreement score.

[7] We take the average to obtain scalar value between $[-1., 1.]$ for each aspect of a verb's connotation frame. For simplicity, we cutoff the ranges of negative, neutral and positive polarities as $[-1, -0.25)$, $[-0.25, 0.25]$ and $(0.25, 1]$, respectively.

Aspect	Algorithm	Acc.	Avg F_1	Aspect	Algorithm	Acc.	Avg F_1
	Majority	56.52	24.07		Majority	79.60	29.55
	Graph Prop	59.53	50.20		Graph Prop	71.91	35.10
$\mathcal{P}(w \rightarrow t)$	3-nn	62.88	47.93	$\mathcal{V}(t)$	3-nn	76.25	39.09
	Aspect-Level	67.56	56.18		Aspect-Level	75.92	45.45
	Frame-Level	67.56	56.18		Frame-Level	76.25	48.13
	Majority	49.83	22.17		Majority	89.30	31.45
	Graph Prop	52.84	42.93		Graph Prop	84.62	38.82
$\mathcal{P}(w \rightarrow a)$	3-nn	55.18	45.88	$\mathcal{V}(a)$	3-nn	85.62	38.45
	Aspect-Level	60.54	60.72		Aspect-Level	87.96	48.06
	Frame-Level	61.87	63.07		Frame-Level	87.96	48.06
	Majority	49.83	22.17		Majority	71.91	27.89
	Graph Prop	52.17	46.57		Graph Prop	69.90	55.57
$\mathcal{P}(a \rightarrow t)$	3-nn	56.52	52.94	$\mathcal{S}(t)$	3-nn	72.91	59.26
	Aspect-Level	63.21	61.70		Aspect-Level	81.61	72.85
	Frame-Level	63.88	62.56		Frame-Level	81.61	72.85
	Majority	48.83	21.87		Majority	50.84	22.47
	Graph Prop	54.85	51.40		Graph Prop	48.83	35.40
$\mathcal{E}(t)$	3-nn	55.18	51.53	$\mathcal{S}(a)$	3-nn	54.85	45.51
	Aspect-Level	64.21	63.63		Aspect-Level	61.54	53.88
	Frame-Level	65.22	64.67		Frame-Level	61.54	53.88
	Majority	49.83	22.17				
	Graph Prop	52.17	35.56				
$\mathcal{E}(a)$	3-nn	54.85	42.63				
	Aspect-Level	62.54	53.82				
	Frame-Level	63.88	56.81				

Table 4.5: Detailed breakdown of results on the development set using accuracy and average F1 over the three class labels (+,-,=).

In the most generic case, the writer is using "contaminate" to frame the agent as being worthless (and even harmful) with regards to the other event participants. For example, in the sentence "his touch contaminated the food," it is clear that the writer presupposes "his touch" to be of negative value in how it impacts the rest of the event.

4.5.2 Connotation Frame Prediction

Using the crowdsourced labels, we randomly divide the annotated verbs into training, dev, and held-out test sets of equal size (300 verbs each). For evaluation we measure average accuracy and F1 score of induced labels for the 9 different connotation frame relationship types for which we have annotations: $\mathcal{P}(w \rightarrow t)$, $\mathcal{P}(w \rightarrow a)$, $\mathcal{P}(a \rightarrow t)$, $\mathcal{V}(t)$, $\mathcal{V}(a)$, $\mathcal{E}(t)$, $\mathcal{E}(a)$, $\mathcal{S}(t)$, and $\mathcal{S}(a)$, where w refers to the writer, a to the agent, and t to the theme.

Baselines To show the non-trivial challenge of learning Connotation Frames, we include a simple majority-class baselines. The MAJORITY classifier assigns each of the 9 relationships the label of the majority of that relationship type found in the training data. Some of these relationships (in particular, the Value of agent/theme) have skewed distributions, so we expect this classifier to achieve a much higher accuracy than random but a much lower overall F1 score.

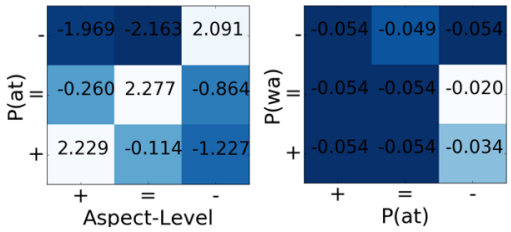
Additionally, we add a GRAPH PROP baseline that is comparable to algorithms like graph propagation or label propagation which are often used for (sentiment) lexicon induction (Velikovich et al., 2010). We use a factor graph with nodes representing the polarity of each typed relation for each verb. Binary factors connect nodes representing a particular type of relation for two similar verbs (e.g. $\mathcal{P}(w \rightarrow t)$ for verbs *persuade* and *convince*). These binary factors have hand-tuned potentials that are proportional to the cosine similarity of the verbs' embeddings, encouraging similar verbs to have the same polarity for the various relational aspects. We use words in the training data as the seed set and use loopy belief propagation to propagate polarities from known nodes to the unknown relationships.

Finally, we use a 3-NEAREST NEIGHBOR baseline that labels relationships for a verb based on the predicate's 300-dimensional word embedding representation, using the same embeddings as in our aspect-level. 3-NEAREST NEIGHBOR labels each verb using the polarities of the three closest verbs found in the training set. The most similar verbs are determined using the cosine similarity between word embeddings.

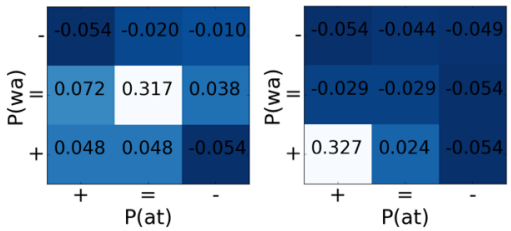
Results As shown in Table 4.5, aspect-level and frame-level models consistently outperform all three baselines — MAJORITY, 3-NN, GRAPH PROP in the development set across the different types of relationships. In particular, the improved F1 scores show that these models are able to perform better across all three classes of labels even in the most skewed cases. The frame-level model also frequently improves the F1 scores of the labels from what they were in the aspect-level model. The summarized comparison of the classifiers' performance test set is shown in Table 4.6. As with the development set, aspect-level and frame-level are both able to outperform the baselines. Furthermore, the frame-level formulation is

Algorithm	**Acc.**	**Avg F_1**
Graph Prop	58.81	41.46
3-nn	63.71	47.30
Aspect-Level	67.93	53.17
Frame-Level	68.26	53.50

Table 4.6: Performance on the test set. Results are averaged over the different aspects.



(a) $\theta_{\mathcal{P}(a \to t)}$ **(b)** θ_{PT}, $\mathcal{P}(w \to t)$: $-$



(c) θ_{PT}, $\mathcal{P}(w \to t)$: $=$ **(d)** θ_{PT}, $\mathcal{P}(w \to t)$: $+$

Figure 4.3: Learned weights, θ, of embedding factor for the perspective of agent to theme (4.3a) and the weights the perspective triad (PT) factor (4.3b-4.3d). Lighter shades are for weights that are more positive, whereas dark blue is more negative.

able to make improvement over the results of the aspect-level classification, indicating that the modelling of inter-dependencies between relationships did help correct some of the mistakes made.

One point of interest about the frame-level results is whether the learned weights over the consistency factors match our initial intuitions about inter-dependencies between relationships. The weights learned in our algorithm can tell us the degree to which these inter-dependencies are actually found in the annotated data.

We show the heat maps for some of the learned weights in Figure 4.3. In 4.3a, we show the weights of one of the embedding factors, and how the node's polarities are more strongly weighted when they match the aspect-level output. In the rest of the figure, we show the weights for the other perspective relationships when $\mathcal{P}(w \to t)$ is negative (4.3b), neutral (4.3c), and positive (4.3d), respectively. Based on the expected interdependencies, when $\mathcal{P}(w \to t) : -$, the model should favor $\mathcal{P}(w \to a) \neq \mathcal{P}(a \to t)$ and when $\mathcal{P}(w \to t) : +$, the model should favor $\mathcal{P}(w \to a) = \mathcal{P}(a \to t)$. Our model does, in fact, learn a similar trend, with slightly higher weights along these two diagonals in the maps 4.3b and 4.3d. Interestingly, when $\mathcal{P}(w \to t)$ is neutral, weights slightly prefer for the other two perspectives to resemble one another, but with highest weights being when other perspectives are also neutral.

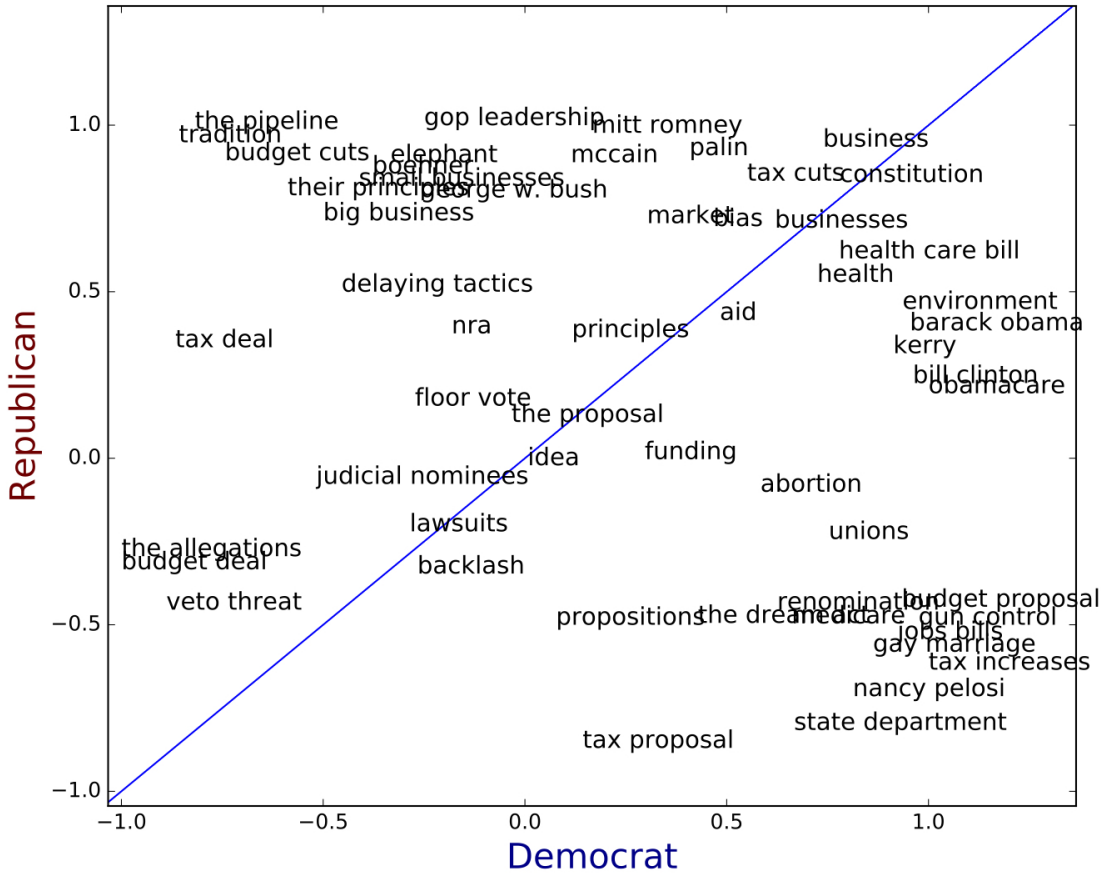


Figure 4.4: Average sentiment of Democrats and Republicans (as agents) to selected nouns (as their themes), aggregated over a large corpus using the learned lexicon (§4.5.2). The line indicates identical sentiments, i.e. Republicans are more positive towards the nouns that are above the line.

4.5.3 Analysis of a Large News Corpus

Using the connotation frame, we present measured implied sentiment in online journalism.

Data From the Stream Corpus (KBA, 2014), we select 70 million news articles. We extract subject-verb-object relations for this subset using the direct dependencies between noun phrases and verbs as identified by the BBN Serif system, obtaining 1.2 billion unique tuples of the form *(url,subject,verb,object,count)*. We also extract tuples from news articles from the Annotated English Gigaword Corpus (Napoles et al., 2012), which contains nearly 10 million articles, resulting in an additional 120 million unique tuples.

Estimating Entity Polarities Using connotation frames, we can also measure entity-to-entity sentiment at a large scale. Figure 4.4, for example, presents the polarity of entities "Democrats" and "Republicans" towards a selected set of nouns, by computing the average estimated $\mathcal{P}(a \rightarrow t)$ polarity (using our frame-level output) over triples where one of these entities appears as part of the phrase in the agent role (e.g. "Democrats" or "Republican party"). There are some nouns towards which both entities are positive ("business", "constitution") or negative ("the allegations","veto threat"). However, we can also see interesting examples in which Democrats feel more positively (below the line: "nancy pelosi", "unions", "gun control", etc.) and ones where Republicans feel more positive ("the pipeline", "gop leadership", "budget cuts", etc.). Also, both entities are neutral towards "idea" and "the proposal", which probably owes to the fact that ideas or proposals can be good or bad for either entity depending on the context.

4.6 Summary

We present a novel formalism of connotative frames that define a set of implied sentiment and presupposed facts for a predicate. We also empirically explore different methods of inducing and modelling these connotation frames, incorporating the interplay between relations within frames. Our work suggests new research avenues for using connotation frames in tasks that require deeper understanding of social and political discourse. In future work, connotation frames could be used to detect writer's perspective and extrapolate a more complex understanding of the intent of a given article. By having more insights about the connotation of a piece of text, we may be able to automatically detect persuasive language, biased text, or misleading information.

Chapter 5

Generating Empathetic Dialogue Responses

In this chapter, we explore social reasoning about a speaker in the specific task of open-domain dialogue. For humans, typical conversations involve reasoning about our conversation partner and their mental state. When someone shares a personal story, we know how to reply in a way that acknowledges that person's feelings. This is a crucial communicative skill when interacting with others, but it is not built into most typical dialogue systems. We discuss a new task for evaluating how well state-of-the-art dialogue models can perform at empathetic response generation. We also present a new conversation dataset and show that this data can be used to train dialogue models. We find that models trained with this data are judged to be more empathetic by human raters. This chapter is based on work originally published in Rashkin et al. (2019).

5.1 Introduction

We explore reasoning over a single entity's mental state, such as when talking to a conversation partner. One desirable trait in a dialogue agent is to appropriately respond when an interlocutor is describing personal experiences, by understanding and acknowledging any implied feelings — a skill we refer to as empathetic responding. For instance, when the conversation partner shares a personal story like "I finally got promoted today", it would be natural to say something that acknowledges their accomplishment like "Congrats! That's great!".

Previous work indicates that this may be an important component for human interactions, showing, for

Label: Afraid
Situation: Speaker felt this when...
"I've been hearing noises around the house at night"
Conversation:
Speaker: I've been hearing some strange noises around the house at night.
Listener: oh no! That's scary! What do you think it is?
Speaker: I don't know, that's what's making me anxious.
Listener: I'm sorry to hear that. I wish I could help you figure it out

Label: Proud
Situation: Speaker felt this when...
"I finally got that promotion at work! I have tried so hard for so long to get it!"
Conversation:
Speaker: I finally got promoted today at work!
Listener: Congrats! That's great!
Speaker: Thank you! I've been trying to get it for a while now!
Listener: That is quite an accomplishment and you should be proud!

Figure 5.1: Two examples from EMPATHETICDIALOGUES training set. The first crowdsource worker (the speaker) is given an emotion label and writes their own description of a situation when they've felt that way. Then, the speaker tells their story in a conversation with a second worker (the listener).

example, that humans are often more efficient when engaging in social talk with each other (Wentzel, 1997; Levinson et al., 2000; Bickmore and Cassell, 2001; Kim et al., 2004; Fraser et al., 2018) and that people may attempt to engage automated systems in a similarly social way (Reeves and Nass, 1996; Lee et al., 2010). In more applied settings, such as call centers for customer or health services, it is important to have dialogue agents that are able to respond empathetically for better callers' experiences (Clark et al., 2013).

To facilitate evaluation of a model's ability to produce empathetic responses, we introduce a new task for dialogue systems to respond to people discussing situations that cover a wide range of emotions, and EMPATHETICDIALOGUES (ED), a novel dataset with about 25k personal dialogues. Each dialogue is grounded in a specific situation where a speaker was feeling a given emotion, with a listener responding (as in Figure 5.1). This dataset is larger and contains a more extensive set of emotions than many similar emotion prediction datasets from other text domains, such as Scherer and Wallbott (1994), Strapparava and Mihalcea (2007), Mohammad et al. (2018), and Gupta et al. (2017).

Our experiments show that commonly used conversation models trained on internet conversation data are not rated as very empathetic. We propose two simple ways to leverage our dataset to improve those models: use utterances from our training data as candidate responses in a retrieval model at inference time, and fine-tune the model on our task. Finally, we explore whether different ways of combining information from external predictors can lead to more empathetic responses.

5.2 Related Work

Emotion Prediction There is a wide breadth of research in emotion classification tasks (Duppada et al., 2018; Park et al., 2018; Xu et al., 2018b; Mohammad et al., 2018) that build on deep networks pretrained on large-scale weakly-labelled data such as emojis (Felbo et al., 2017) or hashtags (Mohammad, 2012), gathered from public social media content published on Twitter. The SEMEVAL2019 EmoContext challenge also uses conversation data for detection of three basic emotions ('happy', 'sad', and 'angry') over two turns of context from Twitter exchanges (Gupta et al., 2017). We focus on personal conversations rather than using social media data to be closer to a context of a one-on-one conversation.

Another interesting resource is the DAILYDIALOG (DD) dataset (Li et al., 2017), which comprises about 13k dialogues obtained by crawling educational websites intended for learners of English and also has emotion label annotations. Many of the dialogues are focused on topics for ESL learners (ordering from a restaurant, asking for directions, introductions, etc), but only $\approx 5\%$ of the utterances have a label other than "none" or "happy". Our task focuses explicitly on conversations about emotionally grounded personal situations, and considers a richer, evenly distributed set of emotions. We also introduce an explicit single *listener* in the conversation who is reacting to the situation being described in an empathetic way, to make the setting as close as possible to our desired goal of a one-on-one empathetic conversation.

Controllable Language Generation Several other works have focused on controlling the emotional content of a text response either through a manually specified target (Zhou and Wang, 2018; Zhou et al., 2018b; Wang and Wan, 2018; Hu et al., 2017) or through a general term to encourage higher levels of affect (Asghar et al., 2018) or politeness (Niu and Bansal, 2018), with evaluations focused on matching a predetermined desired emotion. Similar to our work, some of these investigations (Huang et al., 2018) have focused on generating dialogue utterances for a specific emotion that is chosen manually. Our work focuses on empathetic responses that are appropriate to signals inferred purely from text rather than conveying a pre-specified emotion.

5.3 Talking about Personal Situations

We consider an open-domain one-on-one conversational setting where two people are discussing a situation that happened to one of them, related to a given feeling. We collect around 25k conversations using the following format.

Emotional Situation Grounding Each conversation is grounded in a situation, which one participant writes about based on a given emotion label. We consider 32 emotion labels, listed in Figure 5.2, which we chose by aggregating labels from several emotion prediction datasets (Scherer and Wallbott, 1994; Strapparava and Mihalcea, 2007; Skerry and Saxe, 2015; Li et al., 2017; Mohammad, 2012). These emotion labels cover a broad range of positive and negative emotions. Our goal in providing a single emotion label is to have a situation strongly related to (at least) one particular emotional experience, though we note that some emotions may be very closely related and additional related emotions may be invoked in a given conversation.

Emotion	Most-used speaker words	Most-used listener words	Training set emotion distrib
Surprised	got,shocked,really	that's,good,nice	5.1%
Excited	going,wait,i'm	that's,fun,like	3.8%
Angry	mad,someone,got	oh,would,that's	3.6%
Proud	got,happy,really	that's,great,good	3.5%
Sad	really,away,get	sorry,oh,hear	3.4%
Annoyed	get,work,really	that's,oh,get	3.4%
Grateful	really,thankful,i'm	that's,good,nice	3.3%
Lonely	alone,friends,i'm	i'm,sorry,that's	3.3%
Afraid	scared,i'm,night	oh,scary,that's	3.2%
Terrified	scared,night,i'm	oh,that's,would	3.2%
Guilty	bad,feel,felt	oh,that's,feel	3.2%
Impressed	really,good,got	that's,good,like	3.2%
Disgusted	gross,really,saw	oh,that's,would	3.2%
Hopeful	i'm,get,really	hope,good,that's	3.2%
Confident	going,i'm,really	good,that's,great	3.2%
Furious	mad,car,someone	oh,that's,get	3.1%
Anxious	i'm,nervous,going	oh,good,hope	3.1%
Anticipating	wait,i'm,going	sounds,good,hope	3.1%
Joyful	happy,got,i'm	that's,good,great	3.1%
Nostalgic	old,back,really	good,like,time	3.1%
Disappointed	get,really,work	oh,that's,sorry	3.1%
Prepared	ready,i'm,going	good,that's,like	3%
Jealous	friend,got,get	get,that's,oh	3%
Content	i'm,life,happy	good,that's,great	2.9%
Devastated	got,really,sad	sorry,oh,hear	2.9%
Embarrassed	day,work,got	oh,that's,i'm	2.9%
Caring	care,really,taking	that's,good,nice	2.7%
Sentimental	old,really,time	that's,oh,like	2.7%
Trusting	friend,trust,know	good,that's,like	2.6%
Ashamed	feel,bad,felt	oh,that's,i'm	2.5%
Apprehensive	i'm,nervous,really	oh,good,well	2.4%
Faithful	i'm,would,years	good,that's,like	1.9%

Figure 5.2: Distribution of conversation labels within training set and top 3 content words used by speaker/listener per category.

Speaker and Listener The person who wrote the situation description (*Speaker*) initiates a conversation to talk about it. The other conversation participant (*Listener*) is not given information about the prompt. Rather they must learn about the underlying situation naturally through the conversation with the Speaker. Each conversation lasts for at least four dialogue turns and at most eight. We include two example conversations from the training data in Figure 5.1. The models discussed below are tested in the role of *Listener* responding to the Speaker. Neither the situation description written by the Speaker nor the emotion label is given to the models (just as they were not given to the Listener during dialogue collection). For future work, our data could also be used to generate conversations for the Speaker conditioned on the situation description as an

investigation of how people discuss personal stories.

Collection Details We collected crowdsourced dialogues using the ParlAI platform (Miller et al., 2017) to interact with Amazon Mechanical Turk (MTurk), hiring 810 US workers. A pair of workers are asked to (i) select an emotion word each and describe a situation when they felt that way, and to (ii) have a conversation about each of the situations, as outlined below. Each worker had to contribute at least one situation description and one pair of conversations: one as Speaker about the situation they contributed, and one as Listener about the situation contributed by another worker. They were allowed to participate in as many hits as they wanted for the first $\sim$10k conversations, then we limited the more "frequently active" workers to a maximum of 100 conversations. The median number of conversations per worker was 8, while the average was 61 (some workers were more active contributors than others). To ensure quality, we manually checked random subsets of conversations by our most-frequent workers.

Task Set-up In the first stage of the task, workers are asked to describe in a few sentences a situation based on a feeling label. We ask the workers to try to keep these descriptions between 1-3 sentences. In the second stage, two workers are paired and asked to have two short chats with each other. In each chat, one worker (*Speaker*) starts a conversation about the situation they previously described, and the other worker (*Listener*) responds. Neither can see what the other worker was given as emotion label or the situation description they submitted, so they must respond to each other's stories based solely on cues within the conversation. Each conversation is allowed to be 4-8 utterances long (the average is 4.31 utterances per conversation). The average utterance length was 15.2 words long.

EMPATHETICDIALOGUES Dataset Statistics The resulting dataset comprises 24,850 conversations about a situation description, gathered from 810 different participants. We split the conversations into approximately 80% train, 10% validation, and 10% test partitions. To prevent overlap of discussed situations between partitions, we split the data so that all sets of conversations with the same speaker providing the initial situation description would be in the same partition. The final train/val/test split was 19533 / 2770 / 2547 conversations, respectively. As shown in Figure 5.2, the distribution of emotion label prompts is close to evenly distributed, with a few that are selected slightly more/less often.

5.4 Empathetic Response Generation

This section shows how ED can be used as a benchmark to gauge the ability of a model to respond in an empathetic way, and as a training resource to make generic chitchat models more empathetic. We also examine different ways existing models can be combined to produce more empathetic responses. We use ED dialogues to train and evaluate models in the task of generating conversation responses in the *Listener* role. To emulate a normal conversation, the model has access to previous utterances in the dialogue, but not to the emotion word prompt (e.g., "proud"), nor to the situation description generated by the Speaker. Given a dialogue context x of n previous conversation utterances concatenated and tokenized as $x_1, \cdots, x_m$, followed by a target response $\bar{y}$, our models are trained to maximize the likelihood $p(\bar{y}|x)$ of producing the target response. We investigate two common settings for dialogue agents: generation-based and retrieval-based (Lowe et al., 2016) as described in Figure 5.3.

5.4.1 Base Architecture

We base our models on Transformer networks (Vaswani et al., 2017), which have proven successful in machine translation and dialogue generation tasks (Zhang et al., 2018; Mazare et al., 2018).

Retrieval-based In the retrieval-based set-up, the model is given a large set Y of candidate responses and picks the "best" one, y^*. We first experiment with the retrieval Transformer-based architecture from Yang et al. (2018): two Transformer encoders separately embedding the context, x, and candidates, $y \in Y$, as h_x and h_y, respectively. We also experiment with BERT (Devlin et al., 2018) as a base architecture to encode candidates and contexts, using the final hidden vector from BERT as the h_x

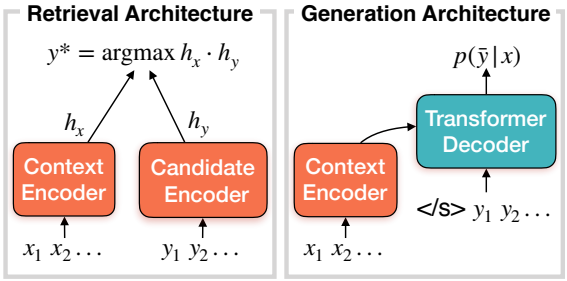


Figure 5.3: Dialogue generation architectures used in our experiments. The context of concatenated previous utterances is tokenized into $x_1, x_2, \cdots$, and encoded into vector h_x by the context encoder. *Left:* In the retrieval set-up, each candidate y is tokenized into $y_1, y_2, \cdots$ and encoded into vector h_y by the candidate encoder. The system outputs the candidate y^* that maximizes dot product $h_x \cdot h_y$. *Right:* In the generation set-up, the encoded context h_x is used as input to the decoder to generate start symbol </s> and tokens $y_1, y_2, \cdots$. The model is trained to minimize the negative log-likelihood of target sequence $\bar{y}$ conditioned on context.

or h_y encodings. Models in the retrieval set-up choose a candidate utterance according to a softmax on the dot product: $h_x \cdot h_y$. We minimize the negative log-likelihood of selecting the correct candidate. At training time, we use all of the utterances from the batch as candidates, with a large batch size of 512 to give the model more negative examples (except for BERT for which a batch size of 256 was used). At inference time, we experiment with two sets of candidate utterances for the model to choose from: all of the response utterances in the ED training set (Y^{ED}) and a million utterances from a dump of 1.7 billion Reddit (R) conversations (Y^{R}).

Generation In the generation set-up, we use the full Transformer architecture (Vaswani et al., 2017), consisting of an encoder and a decoder. The Transformer decoder uses the encoder output to predict a sequence of words y, and is trained to minimize the negative log-likelihood of the target sequence $\bar{y}$. At inference time, we use diverse beam search from Vijayakumar et al. (2016).

Training Details Models are pretrained on predicting replies from a dump of 1.7 billion Reddit conversations, starting either from scratch for the Transformer architectures, or from the BERT$_{base}$ model released by Devlin et al. (2018) for the BERT-based architectures. Pretrained models without any fine-tuning on ED will be referred to as "Pretrained" hereafter. We limit the maximum number of word tokens in the context and response to be 100 each. The Transformer networks used in most experiments have the same base architecture (four layers and six transformer heads) and are trained the same way as in Mazare et al. (2018). For all models, we keep the version that has the lowest loss on the validation set. For the Transformer models, we use 300-d word embeddings pretrained on common-crawl data using fastText (Grave et al., 2018), and for the BERT models, we use 768-d BPE embeddings as pretrained on BooksCorpus and English Wikipedia (Devlin et al., 2018).

5.4.2 Leveraging the Training Data from ED

A retrieval-based model relies on candidates. ED data was explicitly collected with instructions to be empathetic, in a one-on-one setting, which is not the case of the Reddit conversation data used for pretraining, and these domain candidates may be better suited to empathetic responding than generic conversation utterances. Thus, we experiment with incorporating ED training candidates into the pool used at inference

time by pretrained retrieval-based models, with no fine-tuning on ED. For retrieval-based and generation models, we also experiment with fine-tuning pretrained models to predict the next utterance over ED with a context window of four previous utterances, which is the average length of a conversation in our dataset. These models are referred to as "Fine-Tuned" models. This fine-tuning is conducted until convergence for all architectures except those referred to as "Pretrained".

5.4.3 Adding Information from External Predictors

Many existing models have been pretrained on supervised tasks that may be relevant to empathetic responding. We experiment with adding supervised information from two prediction tasks: emotion detection, which is more closely relevant to our task, and topic detection, which may also be useful in crafting relevant replies.

Prepending Top Predicted Labels This set-up (Fig. 5.4), PREPEND, is a very simple way to add supervised information to data, requires no architecture modification, and can be used with black-box classifiers. The top predicted label from the supervised classifier is merely prepended to the beginning of the token sequence as encoder input, as below:

Original:"I finally got promoted!"

Prepend:"`proud` I finally got promoted!"

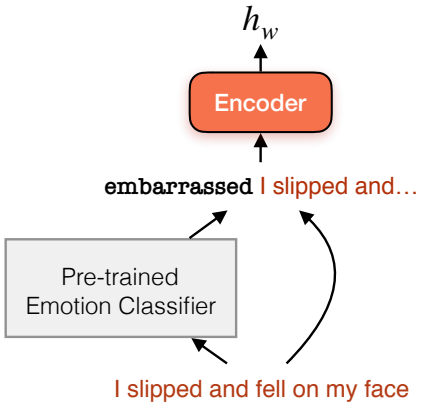


Figure 5.4: Incorporating additional supervised information, here from an emotion classification task. An input sequence (either a dialogue context or a candidate) is run through a pre-trained classifier, and the top output label is prepended to the sequence.

Similar methods have been used for controlling the style of generated text (e.g. Niu and Bansal (2018)). Here, we use a fastText model (Joulin et al., 2017) as prediction architecture. Both the context and the candidates are run through the classifier and receive prepended labels. Fine-tuning is conducted similarly as before, but using these modified inputs. We use two external sources of information. To provide emotion signal, we train a classifier to predict the emotion label from the description of the situation written by the Speaker before the dialogue for the training set dialogues of ED (EMOPREPEND). To gauge whether supervision from a more distant task would still be

	Retrieval			Retrieval w/ BERT		Generation	
Model	Candidate Source	P@1,100	AVG BLEU	P@1,100	AVG BLEU	PPL	AVG BLEU
Pretrained	R	-	4.10	-	4.26	27.96	5.01
	ED	43.25	5.51	49.94	5.97	-	-
Fine-Tuned	ED	**56.90**	5.88	65.92	**6.21**	21.24	**6.27**
EmoPrepend	ED	56.31	5.93	**66.04**	6.20	24.30	4.36
TopicPrepend	ED	56.38	**6.00**	65.96	6.18	**25.40**	4.17

Table 5.1: Automatic evaluation metrics on the test set. Pretrained: model pretrained on a dump of 1.7 billion REDDIT conversations (4-layer Transformer architecture, except when specified BERT). Fine-Tuned: model fine-tuned over the EMPATHETICDIALOGUES training data (Sec. 5.4.2). EmoPrepend, Topic-Prepend: model incorporating supervised information from an external classifiers, as described in Sec. 5.4.3. Candidates come from REDDIT (R), EMPATHETICDIALOGUES (ED), or DAILYDIALOG (DD). P@1,100: precision retrieving the correct test candidate out of 100 test candidates. AVG BLEU: average of BLEU-1,-2,-3,-4. PPL: perplexity. *Bold: best performance for that architecture.*

helpful, we also experiment with a classifier trained on the 20-Newsgroup dataset (Joachims, 1996), for topic classification (TOPICPREPEND).

5.5 Experimental Evaluation

We evaluate the models on their ability to reproduce the Listener's portion of the conversation (i.e. the ability to react to someone else's story). We use both automated metrics and human evaluation to score each model's retrievals/generations. Human evaluation is important, as automated metrics don't always correlate with human judgments of dialogue quality (Liu et al., 2016), but we provide automated metrics to give a sense of how well they align with human judgment on this task.

Automated Metrics (Table 5.1) For both retrieval and generation systems, we compute BLEU scores (Papineni et al., 2002) for the model response, comparing against the gold label (the actual response), following the practice of earlier work in dialogue generation (Wen et al., 2015; Li et al., 2016a,b). For the generation systems, we additionally report perplexity of the actual gold response. For the retrieval-based systems, we further compute `p@1,100`, the accuracy of the model at choosing the correct response out of a hundred randomly selected examples in the test set. When we compute `p@1,100`, the actual response is included

	Model	Candidate	Empathy	Relevance	Fluency
Retrieval	*Pre-trained*	R	2.82 ± 0.12	3.03 ± 0.13	4.14 ± 0.10
		ED	3.45 ± 0.12	3.55 ± 0.13	4.47 ± 0.08
	Fine-tuned	ED	$\mathbf{3.76 \pm 0.11}$	3.76 ± 0.12	4.37 ± 0.09
	EmoPrepend	ED	3.44 ± 0.11	3.70 ± 0.11	4.40 ± 0.08
	TopicPrepend	ED	3.72 ± 0.12	$\mathbf{3.91 \pm 0.11}$	$\mathbf{4.57 \pm 0.07}$
Retrieval w/ BERT	*Pre-trained*	R	3.06 ± 0.13	3.29 ± 0.13	4.20 ± 0.10
		ED	3.43 ± 0.13	3.49 ± 0.14	4.37 ± 0.10
	Fine-tuned	ED	3.71 ± 0.12	3.76 ± 0.12	4.58 ± 0.06
	EmoPrepend	ED	3.93 ± 0.12	3.96 ± 0.13	4.54 ± 0.09
	TopicPrepend	ED	$\mathbf{4.03 \pm 0.10}$	$\mathbf{3.98 \pm 0.11}$	$\mathbf{4.65 \pm 0.07}$
Generation	*Pre-trained*	–	2.31 ± 0.12	2.21 ± 0.11	3.89 ± 0.12
	Fine-Tuned	–	$\mathbf{3.25 \pm 0.12}$	$\mathbf{3.33 \pm 0.12}$	4.30 ± 0.09
	EmoPrepend	–	3.16 ± 0.12	3.19 ± 0.13	4.36 ± 0.09
	TopicPrepend	–	3.09 ± 0.13	3.12 ± 0.13	$\mathbf{4.41 \pm 0.08}$
Gold Response	–	–	4.19 ± 0.10	4.55 ± 0.07	4.68 ± 0.06

Table 5.2: Human ratings. Fine-tuning on ED and using ED candidates generally improves scores, especially on Empathy, with minimal retraining. Additional external supervision (Prepend) improves the Empathy and Relevance scores for BERT-based models. Bold: best score for that group. Italics: reference model for the group.

in the candidates, unlike inference from the retrieval systems for all other metrics, which only uses training utterances as candidates.

Human Ratings (Table 5.2) We ran crowdsourcing tasks using MTurk. Participants were given a model's output for a randomly selected test set example and asked to score different aspects of the model. The rating task provides a means of comparing aspects of responses, and we ask raters specifically about whether the response is acknowledging the conversation partner's feelings. We collected at least 100 ratings per model and asked about three aspects of performance, all rated on a Likert scale (1: not at all, 3: somewhat, 5: very much):

Empathy/Sympathy: did the responses show understanding of the feelings of the person talking about their experience?

Relevance: did the responses seem appropriate to the conversation? Were they on-topic?

Fluency: could you understand the responses? Did the language seem accurate?

Context	Setup	Model	Cand	Generated Response
Someone came to my door with a gun the other day.	Gen	Pretrained	-	I'm not sure if you're joking or not.
	Gen	Fine-Tuned	-	Oh no! What did you do?
	Gen	EmoPrepend	-	Oh no! Did you have a weapon?
	Gen	TopicPrepend	-	I'm glad to hear that.
My son failed his exams! Can you believe it! I don't know what to do!	Ret-BERT	Pretrained	R	Hopefully his mistakes can help you know what not to do on the tests.
	Ret-BERT	Pretrained	ED	Oh no! I ' m so sorry! I hope you could pull your grade up!
	Ret-BERT	Fine-Tuned	ED	Oh no! Did he not study?
	Ret-BERT	EmoPrepend	ED	Im sorry about that! Did your son start school already?
	Ret-BERT	TopicPrepend	ED	Oh no! Does he have the option of taking it again?

Table 5.3: Examples of model responses. Shading is provided for better legibility.

5.6 Results

Pretrained Models Baseline Pretrained conversation models are rated poorly by humans for empathy when the candidates are retrieved from Reddit utterances or when a generation model is used (Table 5.2). Higher ratings with models based on BERT show that larger models may seem more empathetic, but still remain far from human performance.

Using EMPATHETICDIALOGUES for Candidate Selection Table 5.1 shows that merely using the pool of candidates from the training set of ED improves the BLEU scores of retrieval models. Using candidates from our dataset also substantially improves the performance of pre-trained retrieval models on all human metrics, particularly the Empathy subscore of most interest to us (Table 5.2).

Using EMPATHETICDIALOGUES for Fine-tuning Additionally, fine-tuning to predict conversation responses on our data improves all automated metrics (Table 5.1). While fine-tuning on ED data improves performance on predicting the next ED utterance, this may come at the expense of performance when predicting next utterances in other corpora.

To investigate if this is a problem, we compared automatic metrics on next utterance prediction with pretrained models and models fine-tuned using ED data (for our retrieval-based Transformer models) when predicting on the original REDDIT data and DAILYDIALOG (drawing both context and candidates from the corpus that we test on) with results shown in Table 5.4. Fine-tuning on ED leads to a performance increase on DAILYDIALOG in both precision and BLEU. The slight decrease of performance on precision of Reddit predictions is not surprising because the pre-trained model was trained directly using Reddit data. But, the consistent improvement on DAILYDIALOG is an encouraging sign that improvements from fine-tuning on ED may generalize to other conversation datasets. This is also encouraging because the style of conversations in DAILYDIALOG (primarily one-on-one conversations about personal life written for ESL learners) is probably closer to the personal conversations that our task is trying to capture (rather than social media styles of communicating that are more likely to be found in Reddit).

	P @1,100		BLEU	
Model	DD	R	DD	R
Pretrained	39.04	58.95	6.65	1.43
Fine-Tuned	44.58	56.25	7.14	1.64

Table 5.4: Performance of the retrieval-based pretrained model and retrieval-based models fine-tuned on ED data for next utterance prediction in other datasets, with both context and candidates from the same dataset (R=Reddit, DD=DailyDialog).

In terms of human evaluations, fine-tuning on the ED data also generally improves human ratings on the ED task, in both retrieval and generation set-ups (Table 5.2).

Augmenting Conversation Models with External Pretrained Classifiers Automated and human evaluations suggest that prepending emotion or topic predictions may boost perfomance of the BERT-based models (but not the smaller models), with Empathy ratings close to approaching human performance. Our results indicate that further investigation in techniques to add emotion or topic predictors may be a promising direction for future research.

Resources and Capacity Table 5.5 quantifies resource and parameter usage for several models and set-ups, including a larger Transformer generation model (5 layers instead of 4) and BERT-based architectures with substantially more parameters that require longer training. Using ED candidates in pretrained retrieval models, or fine-tuning pretrained conversation models on ED data makes smaller models perform better than

	Model	Params, resources, train examples	Emp	Rel	Fluent
Retrieval	Pretrained-R	84.3M, 2.5 days, 8GPUs, 1.7B	2.8	3.0	4.1
	Pretrained-ED	same , same, same	3.5	3.6	4.5
	Fine-Tuned	same , + 0.5 hour, 1 GPU, +22.3k	3.8	3.8	4.4
	Pretrained-Bert-R	217M, 13.5 days, 8GPUs , 1.7B	3.1	3.3	4.2
	Pretrained-Bert-ED	same, same, same	3.4	3.5	4.4
	Fine-Tuned-Bert	same, +1hour, 8GPUs, +22.3k	3.7	3.8	4.6
Generation	Pretrained	85.1M, 2 days, 32 GPUs, 1.7B	2.3	2.2	3.9
	Fine-Tuned	same , +1 hour, 1 GPU, +22.3k	3.3	3.3	4.3
	Pretrained-Large	86.2M, 2.5 days, 32 GPUs, 1.7B	2.8	3.0	4.0
	Fine-Tuned-Large	same , +0.5 hour, 1 GPU, +22.3k	3.6	3.6	4.5

Table 5.5: Training resources for different models, with human ratings for empathy (Emp), relevance (Rel) and fluency (Fluent). Retrieval-based models use reply candidates from the ED training set (ED) or from Reddit (R). Resource comparisons are relative to the first row of each group. Fine-tuning on ED improves all scores (except for Fluency in one case) while requiring minimal additional training resources. SEM is approximately 0.1

larger ones with minimal increase in resource usage.

5.7 Summary

We introduce a new dataset of 25k dialogues grounded in situations prompted by specific emotion labels. Our experiments show that using this dataset to provide retrieval candidates or fine-tune conversation models leads to responses that are evaluated as more empathetic. Initial results on this dataset also indicate that incorporating external knowledge, such as using external predictors of topic or emotion, may improve slightly on this task. Future work can investigate other methods of incorporating external inferences. Insights from learning on this task may be useful in a variety of open-domain dialogue situations or even in task-oriented dialogues where anticipating a interlocutor's feelings might help lead to smoother, more productive conversations.

Chapter 6

Conclusion

This dissertation investigates approaches for integrating social commonsense reasoning with NLP systems. We present new focused tasks and resources for training machine learning models to learn implicit social implications of text. We also explore new modelling frameworks for integrating social inferences towards downstream NLP tasks such as story or dialogue generation.

First we investigated approaches for reasoning about social dynamics of narratives, looking at the relationships between characters and plot events. In Chapter 2, we described the Story Commonsense dataset, a novel large-scale dataset with mental state annotations of characters in 15k stories. We created a complex annotation pipeline which allowed us to collect motivation and emotion changes as both psychology categories and open-text explanations. We designed two tasks for predicting character's mental state changes and established baseline performance using commonly used NLP models.

Our results show that this is a challenging task for models. Error analysis indicates that one opportunity for future work may involve including external knowledge resources to achieve higher performance, particularly in predicting rare classes. We also showed that context is extremely important to interpretation of character mental state, indicating the need for robust context representations in this task.

We also explored models for tracking plot dynamics while generating stories (Chapter 3). We designed a new task for story generation conditioned on an unordered outline of plot elements. This is a complex task in which a model must interweave plot elements dynamically based on what it's already written. We presented PLOTMACHINES, a novel model for writing stories that uses plot state tracking and high-level

discourse structure to write tighter stories based on outlines.

Next, we explored methods for making inferences about the mental state of the speaker or writer of a natural language utterance. In Chapter 4, we introduced connotation frames, a new formalism for understanding the connotative relations conferred by a predicate to its arguments. We demonstrated that this type of relation can be learned relatively well using computational methods. We also used the created lexicon for large-scale news analysis of stance, as an example of one downstream application of connotative information. The connotation frames lexicon could be extended for more holistic analysis of writer's intent in text. This could have many positive social applications such as creating interactive tools fro media literacy.

Finally, in Chapter 5, we discussed how social inferences about a speaker may be used in conversational settings. We introduced the task of empathetic response generation, in which a dialogue agent must reply appropriately to someone's personal situation. We created EMPATHETICDIALOGUES, a new large scale dataset of 25k conversations covering a diverse set of emotion labels. We explored how well state-of-the-art dialogue models can be adapted to this task. We found that leveraging EMPATHETICDIALOGUES, we can improve the quality of dialogue responses, with humans judging them to be empathetic. We also found that including predictions from external classifiers of topic or emotion may have positive effects on model performance, indicating one direction for future work.

6.1 Future Directions

There are still many different opportunities for future research in this area. The tasks described in this dissertation have been shown to be challenging even for state-of-the-art models. There are also a myriad of types of social inference tasks that are still unexplored. We conclude by discussing a few possible future directions.

Modeling with external commonsense resources One possible avenue for future research is to find new methods for including knowledge from external commonsense resources like CONCEPTNET (Speer et al., 2017), ATOMIC (Sap et al., 2019a; Rashkin et al., 2018b), or COMET (Bosselut et al., 2019). This presents a significant challenge because tuples from these resources are often abstract, generic, or out-of-context. Further research is needed to discover the best way of combining this type of external information with

additional context information in order to integrate with systems for downstream tasks. Integrating information from resources like connotation frames (Chapter 4) may also be beneficial for downstream tasks such as language modeling where we expect the connotation towards certain entities to be described consistently across an entire document.

Modeling Character-centric Social Dynamics In Chapter 3, we presented a new transformer architecture that tracks the states of plot elements from an outline. This architecture can be expanded on for a variety of tasks. One future direction could be to create memory cells that are more character-focused with update rules that are explicitly tied to changes in physical or mental state changes (such as those learned from Chapter 2) that characters undergo throughout a story.

Positive Social Applications Research in this area could have far-reaching implications for designing NLP systems that are able to collaborate with humans in a more productive and natural way. As discussed in Chapter 5, using social commonsense reasoning could be beneficial for dialogue agents to produce more engaging, empathetic, natural conversations. Another positive application of social commonsense reasoning is in helping improve media literacy by creating models for more robust reasoning about writer's intent. Models that generate richer explanations of intent are highly relevant for automatically detecting linguistic bias and subtle persuasive techniques, such as those present in unreliable news sources and propaganda.

Bibliography

Alina Adreevskaia and Sabine Bergler. 2006. Mining wordnet for fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 209–216.

Cem Akkaya, Janyce Wiebe, and Rada Mihalcea. 2009. Subjectivity word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 2, pages 190–199.

Ian Apperly. 2010. *Mindreaders: the cognitive basis of" theory of mind"*. Psychology Press.

Nabiha Asghar, Pascal Poupart, Jesse Hoey, Xin Jiang, and Lili Mou. 2018. Affective neural response generation. In *European Conference on Information Retrieval*, pages 154–166. Springer.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics*, volume 1, pages 86–90.

Steven Bethard, William J Corvey, Sara Klingenstein, and James H Martin. 2008. Building a corpus of temporal-causal structure. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

Timothy Bickmore and Justine Cassell. 2001. Relational agents: a model and implementation of building

user trust. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 396–403. ACM.

EMA Blijd-Hoogewys, PLC Van Geert, M Serra, and RB Minderaa. 2008. Measuring theory of mind in children. psychometric properties of the tom storybooks. *Journal of autism and Developmental Disorders*, 38(10):1907–1930.

Antoine Bosselut, Over Levy, Ari Holtzman, Coin Ennist, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. In *ICLR*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *ACL*.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom Michael Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*.

Center for Media and Democracy. 2012. Sourcewatch: Conservative news outlets. `http://www.sourcewatch.org/index.php/Conservative_news_outlets`.

Center for Media and Democracy. 2013. Sourcewatch: Liberal news outlets. `http://www.sourcewatch.org/index.php/Liberal_news_outlets`.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 2 of *ACL '09*, pages 602–610.

Guan-Lin Chao and Ian Lane. 2019. Bert-dst: Scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer. In *Interspeech*.

Yoonjung Choi and Janyce Wiebe. 2014. +/-effectwordnet: Sense-level lexicon acquisition for opinion inference. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1181–1191. Association for Computational Linguistics.

Colin Mackinnon Clark, Ulrike Marianne Murfett, Priscilla S. Rogers, and Soon Ang. 2013. Is empathy effective for customer service? evidence from call center interactions. *Journal of Business and Technical Communication*, 27(2):123–153.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, 58:92–103.

Lingjia Deng and Janyce Wiebe. 2014. Sentiment propagation via implicature constraints. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Venkatesh Duppada, Royal Jain, and Sushant Hiray. 2018. Seernet at semeval-2018 task 1: Domain adaptation for affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 18–23.

Philip Edmonds and Graeme Hirst. 2002. Near-synonymy and lexical choice. *Computational Linguistics*, 28(2):105–144.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *ACL*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2019a. Strategies for structuring story generation. In *ACL*.

Lisa Fan, Marshall White, Eva Sharma, Ruisi Su, Prafulla Kumar Choubey, Ruihong Huang, and Lu Wang. 2019b. In plain sight: Media bias through the lens of factual reporting. *ArXiv*, abs/1909.02670.

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. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625.

Song Feng, Jun Seok Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 1774–1784. Association for Computational Linguistics.

Charles Fillmore. 1982. Frame semantics. *Linguistics in the morning calm*, pages 111–137.

Charles J. Fillmore. 1976. Frame semantics and the nature of language. In *In Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech, volume 280, pages 20–32.*

Jamie Fraser, Ioannis Papaioannou, and Oliver Lemon. 2018. Spoken conversational ai in video games: Emotional dialogue management increases user engagement. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 179–184. ACM.

Lea Frermann, Ivan Titov, and Manfred Pinkal. 2014. A hierarchical bayesian model for unsupervised induction of script knowledge. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*.

Liye Fu, Jonathan P. Chang, and Cristian Danescu-Niculescu-Mizil. 2019. Inferring advice-seeking intentions from personal narratives: A cloze test formulation. In *NAACL 2019*.

Yoav Goldberg and Jon Orwant. 2013. A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, volume 1, pages 241–247.

Andrew S Gordon and Jerry R Hobbs. 2017. *A Formal Theory of Commonsense Psychology: How People Think People Think*. Cambridge University Press.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge extraction.

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

Amit Goyal, Ellen Riloff, Hal Daumé III, and Nathan Gilbert. 2010b. Toward plot units: Automatic affect state analysis. In *Proceedings of HLT/NAACL Workshop on Computational Approaches to Analysis and Generation of Emotion in Text (CAET)*.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Stephan Greene and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 503–511.

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

David Gunning. 2018. Machine common sense concept paper. *CoRR*, abs/1810.07528.

Umang Gupta, Ankush Chatterjee, Radhakrishnan Srikanth, and Puneet Agrawal. 2017. A sentiment-and-semantics-based approach for emotion detection in textual conversations. *arXiv preprint arXiv:1707.06996*.

Brent Harrison, Christopher Purdy, and Mark O. Riedl. 2017. Toward automated story generation with markov chain monte carlo methods and deep neural networks.

Kazi Saidul Hasan and Vincent Ng. 2013. Frame semantics for stance classification. *Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CONLL)*, pages 124–132.

Fritz Heider. 1946. Attitudes and cognitive organization. *The Journal of psychology*, 21(1):107–112.

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*.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org.

Chenyang Huang, Osmar Zaiane, Amine Trabelsi, and Nouha Dziri. 2018. Automatic dialogue generation with expressed emotions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 49–54.

HWC Library. 2011. Consider the Source: A Resource Guide to Liberal, Conservative, and Nonpartisan Periodicals. `www.ccc.edu/colleges/washington/departments/Documents/PeriodicalsPov.pdf`. Compiled by HWC Librarians in January 2011.

Parag Jain, Priyanka Agrawal, Abhijit Mishra, Mohak Sukhwani, Anirban Laha, and Karthik Sankaranarayanan. 2017. Story generation from sequence of independent short descriptions. *ArXiv*, abs/1707.05501.

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

Thorsten Joachims. 1996. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 427–431.

Nobuhiro Kaji and Masaru Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of html documents. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1075–1083.

Jaap Kamps, Maarten Marx, Robert J Mokken, and Maarten De Rijke. 2004. Using wordnet to measure semantic orientations of adjectives. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation(LREC'04)*, volume 4, pages 1115–1118.

TREC KBA. 2014. Knowledge Base Acceleration Stream Corpus. `http://trec-kba.org/kba-stream-corpus-2014.shtml`.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Chloe Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *EMNLP*.

Sung Soo Kim, Stan Kaplowitz, and Mark V Johnston. 2004. The effects of physician empathy on patient satisfaction and compliance. *Evaluation & the health professions*, 27(3):237–251.

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

Manfred Klenner, Michael Amsler, and Nora Hollenstein. 2014. Verb polarity frames: a new resource and its application in target-specific polarity classification. In *Proceedings of KONVENS 2014*, pages 106–115.

Baris Korkmaz. 2011. Theory of mind and neurodevelopmental disorders of childhood. *Pediatr Res*, 69(5 Pt 2):101R–8R.

George Lakoff. 1993. The contemporary theory of metaphor.

Min Kyung Lee, Sara Kiesler, and Jodi Forlizzi. 2010. Receptionist or information kiosk: how do people talk with a robot? In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pages 31–40. ACM.

Sungjin Lee. 2013. Structured discriminative model for dialog state tracking. In *SIGDIAL*.

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

Douglas B Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

Hector J. Levesque. 2011. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

Hector J Levesque. 2017. *Common sense, the Turing test, and the quest for real AI*. MIT Press.

Wendy Levinson, Rita Gorawara-Bhat, and Jennifer Lamb. 2000. A study of patient clues and physician responses in primary care and surgical settings. *Jama*, 284(8):1021–1027.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 302–308.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 994–1003.

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

Jiwei Li, Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. 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)*, pages 1106–1115, Beijing, China. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 986–995.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics. `https://pypi.org/project/rouge/`.

Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike 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 *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.

Ryan Lowe, Iulian V Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. On the evaluation of dialogue systems with next utterance classification. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 264.

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.

Marie-Catherine de Marneffe, Christopher D. Manning, and Christopher Potts. 2012. Did it happen? the pragmatic complexity of veridicality assessment. *Computational Linguistics*, 38:301–333.

Lara J. Martin, Prithviraj Ammanabrolu, William Hancock, S. Singh, Brent Harrison, and Mark O. Riedl. 2017. Event representations for automated story generation with deep neural nets. In *AAAI*.

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

Pierre-Emmanuel Mazare, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779.

John W. Mccarthy. 1960. Programs with common sense.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. Parlai: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84.

Amy Mitchell, Jeffrey Gottfried, Jocelyn Kiley, and Katerina Eva Matsa. 2014. Political Polarization & Media Habits. `www.journalism.org/2014/10/21/political-polarization-media-habits/`. Produced by Pew Research Center in October, 2014.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *SemEval@NAACL-HLT*.

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

Saif M. Mohammad. 2012. #emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 246–255, Stroudsburg, PA, USA. Association for Computational Linguistics.

Saif M Mohammad and Peter D Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34. Association for Computational Linguistics.

Chris Moore. 2013. *The development of commonsense psychology*. Psychology Press.

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.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics.

Aida Nematzadeh, Kaylee Burns, Erin Grant, Alison Gopnik, and Thomas L. Griffiths. 2018. Evaluating theory of mind in question answering. In *EMNLP*.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics*, 6:373–389.

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

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

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Ji Ho Park, Peng Xu, and Pascale Fung. 2018. Plusemo2vec at semeval-2018 task 1: Exploiting emotion knowledge from emoji and #hashtags. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 264–272.

Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. 2018. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*.

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

Rafael Pérez y Pérez and Mike Sharples. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence*, 13:119 – 139.

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

Julie Porteous and Marc Cavazza. 2009. Controlling narrative generation with planning trajectories: the role of constraints. In *Joint International Conference on Interactive Digital Storytelling*.

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

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018a. Modeling naive psychology of characters in simple commonsense stories. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2289–2299, Melbourne, Australia. Association for Computational Linguistics.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. Plotmachines: Outline-conditioned generation with dynamic plot state tracking. *ArXiv*, abs/2004.14967.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith, and Yejin Choi. 2018b. Event2Mind: Commonsense inference on events, intents, and reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 463–473, Melbourne, Australia. Association for Computational Linguistics.

Hannah Rashkin, Sameer Singh, and Yejin Choi. 2016. Connotation frames: A data-driven investigation. In *ACL*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1650–1659.

Byron Reeves and Clifford Ivar Nass. 1996. *The media equation: How people treat computers, television, and new media like real people and places.* Cambridge university press.

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

Mark O Riedl. 2009. Story planning: Creativity through exploration, retrieval, and analogical transformation. In *Minds and Machines*, volume 20(4):589–614.

Mark O Riedl and Robert Michael Young. 2010. Narrative planning: Balancing plot and character. In *Journal of Artificial Intelligence Research*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

Stuart Rose, Nick Cramer, and Dave Engel. 2010. Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory edited by Michael W. Berry and Jacob Kogan, John Wiley & Sons, Ltd*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *ArXiv*, abs/1907.10641.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In *AAAI*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social IQA: Commonsense reasoning about social interactions. In *EMNLP 2019*.

Roger C Schank and Robert P Abelson. 1975. *Scripts, plans, and knowledge*. Yale University.

Klaus R. Scherer and Harald G. Wallbott. 1994. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66 2:310–28.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers? *ArXiv*, abs/1909.10705.

Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Jianfeng Gao, and Lawrence Carin. 2019. Towards generating long and coherent text with multi-level latent variable models. In *ACL*.

Amy Skerry and Rebecca Saxe. 2015. Neural representations of emotion are organized around abstract event features. *Current Biology*, 25:1945–1954.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *SemEval@ACL*.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *NeurIPS*, pages 2440–2448.

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.

Charles Sutton and Andrew McCallum. 2009. Piecewise training for structured prediction. *Machine Learning*, 77(2):165–194.

Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *NAACL*.

Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 327–335.

Blaise Thomson and Steve J Young. 2010. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech and Languag*, 24(4):562–588.

Paul Trichelair, Ali Emami, Adam Trischler, Kaheer Suleman, and Jackie Chi Kit Cheung. 2019. How reasonable are common-sense reasoning tasks: A case-study on the Winograd schema challenge and SWAG. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3382–3387, Hong Kong, China. Association for Computational Linguistics.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *ArXiv*, abs/1806.02847.

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 *NeurIPS*, pages 5998–6008.

Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 777–785, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424.

Ke Wang and Xiaojun Wan. 2018. Sentigan: generating sentimental texts via mixture adversarial networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4446–4452. AAAI Press.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Kathryn R Wentzel. 1997. Student motivation in middle school: The role of perceived pedagogical caring. *Journal of educational psychology*, 89(3):411.

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*.

Janyce Wiebe and Lingjia Deng. 2014. An account of opinion implicatures. *CoRR*, abs/1404.6491.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.

Michael Wiegand and Josef Ruppenhofer. 2015. Opinion holder and target extraction based on the induction of verbal categories. *Proceedings of the 2015 Conference on Computational Natural Language Learning (CoNLL)*, page 215.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354.

Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. 2018a. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4306–4315, Brussels, Belgium. Association for Computational Linguistics.

Peng Xu, Andrea Madotto, Chien-Sheng Wu, Ji Ho Park, and Pascale Fung. 2018b. Emo2vec: Learning generalized emotion representation by multi-task training. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 292–298.

Yinfei Yang, Steve Yuan, Daniel Cer, Sheng yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning semantic textual similarity from conversations. In *Rep4NLP@ACL*.

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

Tae Yano, Philip Resnik, and Noah A. Smith. 2010. Shedding (a thousand points of) light on biased language. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 152–158.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *AAAI*, pages 7378–7385.

Steve Young, Milica Gasic, Simon Keizer, Francois Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *ArXiv*, abs/1905.12616.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2204–2213.

Deyu Zhou, Linsen Guo, and Yulan He. 2018a. Neural storyline extraction model for storyline generation from news articles. In *NAACL*.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018b. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Xianda Zhou and William Yang Wang. 2018. Mojitalk: Generating emotional responses at scale. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1128–1137.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. *CoRR*, abs/1802.01886.