Story Ending Generation: Exploring Commonsense Reasoning in GPT-2

Erik S. McGuire
erik.s.mcguire@gmail.com

Abstract

Story Ending Generation (SEG) is a Natural Language Generation task that seeks to generate coherent conclusions to brief stories. GPT-2 is a large pretrained causal language model which requires minimal task-specific changes to the architecture for supervised fine-tuning. As it is a relatively new generative model that has outperformed previous state-of-the-art results on a wide range of commonsense reasoning tasks, this project uses GPT-2 to attack the SEG problem by incorporation of an external knowledge graph as well as multi-task fine-tuning on the ROCStories dataset, a corpus of brief stories designed to capture causal and temporal relationships. External knowledge is injected through language modeling on sentences generated from ConceptNet relations. The multi-task fine-tuning setup uses discriminative tasks, namely the Story Cloze Task (SCT) and/or Sentiment Matching (SM), complemented by an auxiliary language modeling objective. A variety of automated evaluation methods are used to compare the endings generated by permutations of the base model engendered by these training regimes. Preliminary results show that the incorporation of ConceptNet via a dataset of 600k samples improves fine-tuning losses, but less so than further fine-tuning with the ROCStories dataset. Evaluations of generation results show little to no benefit for training beyond fine-tuning the base model for the Story Cloze Task.

1. Introduction

Story Ending Generation (SEG) is a recent subtask (Zhao et al., 2018) of story generation whose goal is to generate story endings that are relevant, consistent, and readable. That is, as seen in Table 1, it is a form of conditional generation with story bodies as context, prompting continuations that reference or extend the same entities and events or otherwise deviate minimally from measurable features. Thus, it is a more tractable form of open-ended generation than weaving entire stories out of whole cloth.

While story generation is a demanding task requiring multiple levels of consistency over broad spans of text, tracking agents and actions in various contexts, the more attenuated task of story ending generation begins after such heavy lifting is performed, with the bulk of the data–the main story body for each sample–accessible as rich training information and as grounding for a broad gamut of evaluations of coherence between prompts and continuations.

Story Body:
Javier is feeling thirsty.
He decides to walk to the nearby water fountain.
At the water fountain he drinks some water.
He feels better.
Gold Ending:
Javier is happy that he no longer feels thirsty.
Generated Ending:
He has a good day.

Table 1. Example story prompt of four sentences, with the ground truth ending and an ending generated by a fine-tuned model. Note the generated ending is positive in sentiment and maintains tense and pronoun gender.

Common sense is typically defined as shared, unspoken world knowledge and it is an increasingly prioritized goal in building artificial intelligence systems. In recent years, in the domain of Natural Language Processing (NLP), we have seen a transition (Bowman et al. 2014, Bhagavatula et al. 2019) from logic-based approaches which use formal, artificial representations to derive answers from hand-crafted rules, to Natural Logic (MacCartney and Manning, 2014) approaches which operate on natural language representations, to neural language-based approaches where models trained on natural language data can reproduce logical behavior in solving reasoning tasks over learned vector representations.

These tasks are typically referred to as **Natural Language Inference** (NLI)–determining the truth, neutrality, or contradiction of hypotheses–or **Recognizing Textual Entailment** (RTE): recognizing whether the meaning of a text can be inferred from another Dagan et al. (2005).

For the language-based approach to NLI and RTE, the use of neural network models was popularized with the introduction of the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015), and since then there has been a surge in datasets and shared tasks in the area (Storks et al., 2019).

The recent successes of transfer learning with transformer-based models like GPT-2 for a broad swath of tasks, improving the coeval state-of-the-art with minimal adjustments for fine-tuning (Radford et al. 2019; Storks et al. 2019) has also furthered a shift in focus toward more difficult challenges in NLI and RTE, with commonsense reasoning a prevalent target of deep NLP development (Devlin et al. 2018; Klein and Nabi 2019). This includes more examination of text generation strategies, as the fluent but nonsensical results of GPT-2 (See et al., 2019) has underscored the need to harness strategies which target commonsense reasoning rather than strategies primarily focused on diversity and repetitiveness.

While the research on the purported universality and atomicity of emotion is contentious (Barrett et al., 2019), the putative emotional arcs of narratives are often considered an integral aspect of the experience being transferred to the reader (Reagan et al., 2016) through text, while emotional content is less controversially seen to contribute meaning, such as speakers' attitudes or feelings toward linguistic content (Bender and Lascarides, 2019). The sentiment of plot arcs in the body of a story may be predictive (Zehe et al., 2016) of the sentiment of story endings when such endings are consistent continuations of story plots, such as happy endings for fairy tales after some conflict is resolved, or negative endings for tragedies.

This project attempts to use GPT-2 as a basis for story ending generation by incorporating commonsense knowledge from ConceptNet and modeling sentiment through multi-task fine-tuning on a corpus of stories. *Multi-task learning* (MTL) has become a core component in the development of state-of-the-art deep learning systems. First used in neural NLP by Collobert & Weston (2008, 2011), the aim is to jointly train a network on multiple related tasks in such a way as to create more generalizable representations. *Fine-tuning* pre-trained models involves adding output layers specific to supervised downstream tasks and training these task-specific parameters–as well as the initial pre-trained parameters–on labeled data from these tasks. In the case of multi-task fine-tuning, losses from all tasks are jointly considered when updating parameters. *ConceptNet*[1] (Liu and Singh, 2004) is a semantic network or knowledge graph where each entry is a triple consisting of two entities (nodes) connected by a relation (edge).

In terms of commonsense reasoning, rather than developing models which reason, our aim here is to develop models which leverage patterns in the data left by reasoning human language users in order to construct coherent continuations to brief narrative prompts.

2. Related Work

After its recent introduction, the Story Ending Generation task was first tackled using external commonsense knowledge by Guan et al. (2018), who used *graph attention* (Zhou et al., 2018) to integrate ConceptNet relations, and subsequently by Chen et al. (2019), who used Conceptnet Numberbatch (Speer and Lowry-Duda, 2017), which are word-level embeddings *retrofitted* (Faruqui et al., 2014) with ConceptNet data, whereby pre-trained word vectors were updated with semantic relations from ConceptNet. More recently, Guan et al. (2020) used GPT-2 to generate entire stories using ConceptNet knowledge data, which were directly incorporated by post-training the pre-trained model on an unsupervised language modeling task using sentences created from ConceptNet triples, before supervised fine-tuning. We follow the latter post-training method in attempting to inject ConceptNet knowledge into GPT-2, here for SEG rather than full story generation.

Goel and Singh (2017) were able to outstrip what were then state-of-the-art results on the Story Cloze Task–choosing the correct ending to a story–by selecting endings which matched the average sentiment of the story body, as computed by the VADER[2] (Hutto and Gilbert, 2014) sentiment analyzer. Chen et al. (2019) outperformed the state-of-the-art results on the Story Cloze Task of the time in part by training a model to predict the sentiment of endings as labeled by VADER, in combination with the aforementioned retrofitted ConceptNet embeddings.

Radford et al. (2018) improved on the coeval state-of-the-art on the Story Cloze Task through multi-task fine-tuning. That is, the model was fine-tuned with the joint losses of the Story Cloze Task and an auxiliary language modeling task. The language modeling objective was used to improve generalization and accelerate convergence.

Our work in this project combines the work of Guan et al. (2020) in injecting ConceptNet knowledge into GPT-2 through language modeling on synthetic sentences, adapts the work of Goel and Singh (2017) in the use of sentiment matching, and makes use of multi-task fine-tuning as described by Radford et al. (2018), applied to story ending generation. In so doing, we hope to use a fairly con-

[1] http://conceptnet.io/

[2] https://github.com/cjhutto/vaderSentiment

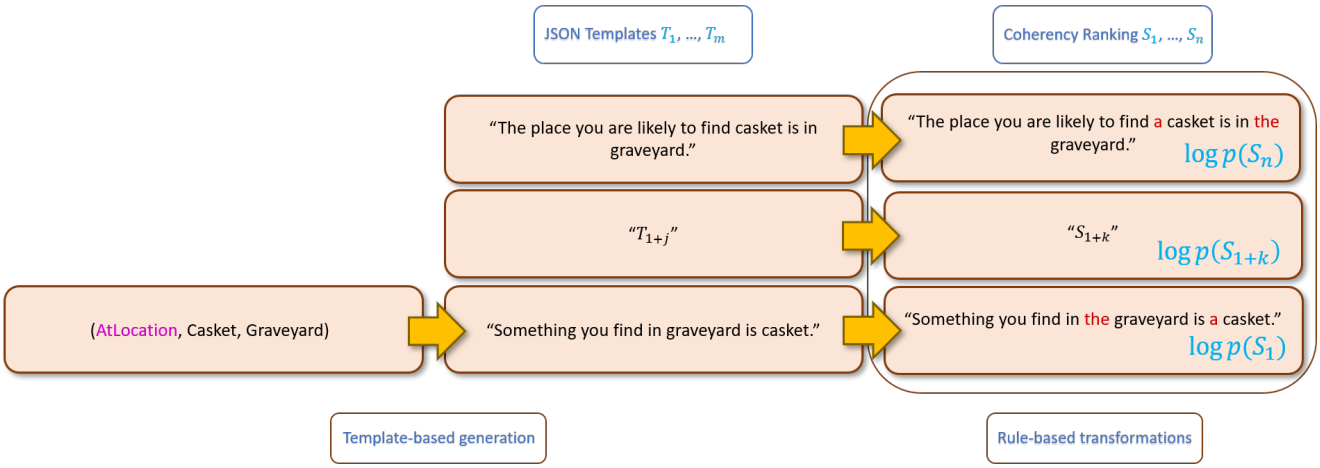

Figure 1. Conversion of a ConceptNet triple into a sentence (Davison et al., 2019). The triple is converted to sentences using JSON templates corresponding to each relation, and transformations of these sentences are ranked for coherency by GPT-2 to select a single representative sentence for the triple.

strained generation task to discern the effects and merits of recent techniques and models used with relative success for commonsense reasoning and story-related tasks.

3. Methodology

The main pipeline of this project is first to train a base GPT-2 model with a language modeling objective on sentences generated from ConceptNet triples, in order to incorporate assertions of commonsense relations between entities by learning to model these relations between entities and their semantic neighbors as probable embedding sequences.

The language modeling objective can be written as follows:

$$\mathcal{L}_{LM} = -\sum_{i} \log P(x_i | x_{i-k}, ..., x_{i-1}; \theta) \qquad (1)$$

where θ are the network parameters and k the context window size (Radford et al., 2018). Optionally, this preliminary language model post-training can be fed on the unlabeled ROCStories dataset rather than or in addition to the ConceptNet training.

Next, multi-task fine-tuning proceeds using the ROCStories dataset and language modeling (LM) as the auxiliary objective. First in this second stage, multi-task fine-tuning is applied to the ConceptNet-trained model using the Sentiment Matching Task as the main task, and then fine-tuned again with the Story Cloze Task as the main task. In each case, the loss is a weighted sum of the main and auxiliary tasks:

$$\mathcal{L}_{MTL} = -\sum_{(x,y)} \log P(y | x^1, ..., x^m) + \lambda \mathcal{L}_{LM} \qquad (2)$$

where y is the label for which ending is correct, x is the story with start, delimiter, and classification tokens, λ is the weighting coefficient, and samples are from the ROCStories dataset used for MTL fine-tuning. Probabilities are determined by softmax, and in the case of Equation 2, the logits or prediction scores for each choice are summaries of the stories' hidden states: that is, they are the hidden states of their respective classification tokens, as described in §3.2.

Other variations of the *Base* → *ConceptNet* → *Sentiment Matching* + *LM* → *SCT* + *LM* pipeline skip the ConceptNet training, or skip the sentiment matching, or reverse the order of the matching and Story Cloze tasks–for example, we may move from the base pre-trained GPT-2 model directly to the multi-task fine-tuning, with the joint Story Cloze and language modeling tasks (*Base* → *SCT* + *LM*).

3.1. ConceptNet

The idea of incorporating knowledge from ConceptNet by converting the relations into sentences appears to be based on *distant supervision* approaches in recent years (Ye et al., 2019)–where sentences which contain the elements of triples are seen to represent those triples–and inspired by the advent of pretrained language models and the desire to directly fine-tune on natural language representations of knowledge triples. For this, as illustrated in Figure 1, we have used template-based code[3] (Davison et al., 2019) which offers a variety of approaches for templates, but in the Coherency Ranking case, generates sentences from triples through hand-crafted tem-

[3]https://github.com/JoshFeldman95/Extracting-CK-from-Large-LM

plates and applies combinations of transformations (e.g., adding articles to nouns) to make them more grammatical and natural, before selecting the best candidate sentence (e.g., "a musician can play a musical instrument") according to GPT-2—the intuition being that ungrammatical sequences (e.g., "musician can playing musical instrument") will have a lower likelihood:

$$S^* = \underset{S \in \mathcal{S}}{\operatorname{argmax}} \left[\log P_{coh}(S) \right] \tag{3}$$

where $\mathcal{S} = \{S_1, ..., S_j\}$ is the set of candidate sentences resulting from template-based generation and rule-based transformations.

3.2. Story Cloze Task

Along with the ROCStories dataset, the Story Cloze Task was introduced by (Mostafazadeh et al., 2016), and involves choosing the correct single sentence ending to a 4-sentence story. The stories are designed to reflect common, everyday scenarios in simple fashion, while containing causal, temporal relationships.

For multi-task fine-tuning, as seen in Figure 2, the stories' training and testing sets are loaded as inputs, where each sample contains two choices: in each case the first four sentences of the story and the respective correct and incorrect endings; and finally, a label corresponding to which version is correct. A special start token is prefixed to each version of the story, with a delimiter token separating the story body from a given ending, and a final classification token is affixed, to be used as the representative hidden state (as opposed to using the hidden states for every token in the story versions' sequences). This hidden state for each version of the story is passed through a linear layer to obtain a single score corresponding to the label, and these two choices' scores are passed through a softmax layer with cross-entropy to obtain the scalar loss. Thus, the loss is the negative log-likelihood (NLL) of the correct choice.

This multiple choice loss is combined as a weighted sum (Eq. 2) with the language modeling loss, which is the NLL for all of the token predictions in both versions of the story, where vocabulary scores are obtained by using a linear layer to project all of the hidden states output by the Transformer's final decoder block onto the token embeddings. Guan et al. (2020) use only the gold stories' LM loss, rather than both correct and incorrect versions, but it is unclear what benefits may be gained or lost by doing so.

The usage of special token embeddings, described by Radford et al. (2018) seems to be informed most recently by Liu et al. (2018)'s decoder-only use of Transformers, with the input and target sequences of a sequence-to-sequence task combined with a separator token and used as a single sequence for language model training (in contrast, at inference, as with conventional teacher forcing methods, only the input sequence is provided, and generated outputs are used autoregressively).

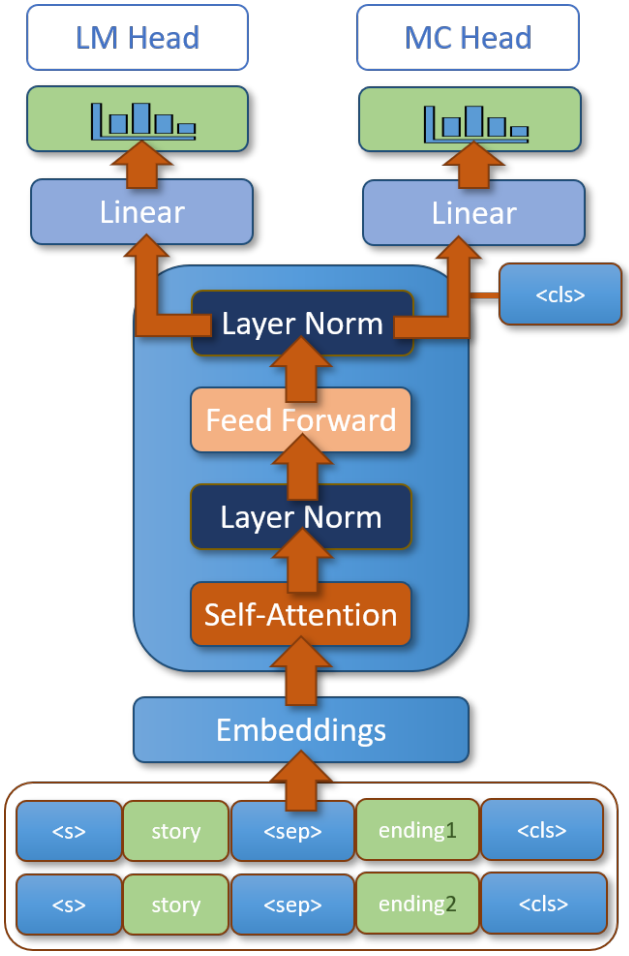

Figure 2. The preprocessing stage constructs a version of each story by concatenating the labeled story with the respective endings, which are delimited from the story by a special token embedding. Both versions are affixed with special classification tokens to extract representations, and in the multiple choice head the vectors for the two choices are projected through a linear layer to obtain prediction scores to normalize with a softmax layer. The preprocessing for the language modeling task uses the same dual versions of the stories, setting the labels to be the same as the inputs (which are shifted to the right elsewhere in the model).

3.3. Sentiment Matching Task

Sentiment matching selects the most congruent ending in terms of the mean composite valence of the story, that is, the ending sentence whose composite score is most similar to the average of the composite scores of the first four sentences.

The stories corpus focuses on the narration of events centering around a single protagonist. This somewhat simplifies the modeling of sentiment in the story over the five sentences, in that the emotional content hews closely to the unfolding of the plot, either through a first-person narrator or third-person description of the character. It seems plausible then that the bulk of changes to a story's sentiment can form a pattern that might allow the model to distinguish or generate affectively congruent endings. Given its usage in various research on stories (Chen et al., 2017), VADER (Hutto and Gilbert, 2014) is used here to annotate the unlabeled portion of the stories dataset: using composite valence scores, following Hutto's suggestions for the thresholds for determining positive, negative, or neutral orientations:

Valence	**Compound Score (s)**
Positive	$s \geq 0.05$
Neutral	$-0.05 < s < 0.05$
Negative	$s \leq -0.05$

Table 2. Heuristic labels for compound scores from VADER annotation.

VADER is a system that uses a combination of a *polarity*-based sentiment lexicon and a small set of heuristics from the results regarding the linguistic cues to changes in *intensity* (e.g., exclamation points, contrastive conjunctions, degree adverbs). Similarity in our implementation is computed simply as the absolute difference between the average of the story body's sentences' compound scores, with ties going to the ground truth ending, such that the correct and distractor endings will be swapped when the correct ending is strictly greater in sentiment distance from the story prompt than the distractor ending is. Thus, there is a level of independence of the Sentiment Matching task from the Story Cloze task, in its effects on model weights.

Ideally, by adjusting weights based on correctly choosing endings closest to the sentiment of a story body, complemented with the auxiliary language modeling task, the model might learn to generate continuations of stories that are more likely to cohere with the story prompts in terms of affect, avoiding jarring dissonance, where perhaps a morose story about injury or illness is abruptly concluded with a very positive non-sequitur.

Recently, inspired by scholarship on neural text degeneration (Holtzman et al., 2019), where argmax approximations such as *beam search* and other maximization approaches lead to 'degenerate' text—especially repetition, HuggingFace[4] recently added options for truncated sampling with softmax *temperature* adjustment and *nucleus sampling* (Figure 3). Softmax temperature adjustments entail 'hardening' the distribution by lowering the temperature from the default 1.0, and thereby decreasing diversity of tokens by creating peaked distributions, or softening probabilities into more uniform distributions by increasing the temperature and thereby making outputs less generic. It is used along with nucleus sampling, a form of top-k (Fan et al., 2018) sampling (sampling from a truncated distribution of k tokens) where the retained tokens to be sampled from is the smallest set whose probability mass surpasses some threshold p.

The code tokenizes the prompt string as a PyTorch tensor using the GPT-2 tokenizer initialized from the pretrained model. It begins with a special start token, generates a sequence until a maximum length or EOS token is attained, decodes the generated tokens, and joins them into readable strings. The subword-level tokenizer is BPE (Gage, 1994), adapted for neural machine translation by (Sennrich et al., 2015) to represent an open vocabulary, by iteratively merging frequent character n-grams; in this way, probabilities may be assigned to any input tokens. For GPT-2, the number of merge operations hyperparameter—which determines the final vocabulary size (the initial size and the character sequence produced by each merge) is 50,000.

The generate function also allows for a *repetition penalty* (Keskar et al., 2019), intended to allow for 'near-greedy' sampling at inference, in order to mitigate repetition: scores for tokens that have been generated previously are reduced by scaling up their softmax temperatures—by default the penalty is 1.2 for tokens in the generated set and 1.0 otherwise. As frequent punctuation was penalized in the original implementation[4], such that every ending concluded with an exclamation point rather than a full stop, here we overrode the methods to exclude punctuation from the penalty.

4. Experiments

4.1. Datasets

In order to train the models in distinct Language Modeling and Multi-Task Fine-Tuning phases, the ROCStories dataset, which comes with training, validation and test splits, was further divided by randomly holding out 5% of the training data and then splitting this equally and randomly into validation and testing sets. The ConceptNet sentences were randomly split with the same proportions. Overlap between ConceptNet vocabulary and ROCStories can be seen in Table 5.

The models are evaluated after generating endings using a held out set of ROCStories unseen during training and validation. As described in Table 4, the test set for generation consists of 2,408 stories.

[4] https://github.com/huggingface/transformers

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	*distinct-1*	*distinct-2*	*distinct-3*	*distinct-4*
Base	0.001	0.000	0.000	0.000	0.023	0.005	0.000	0.974	**0.902**	**0.809**	**0.717**
Base→SCT	**0.072**	**0.038**	**0.030**	**0.027**	0.063	**0.084**	**0.255**	0.999	0.848	0.698	0.550
CN→SCT	0.060	0.029	0.023	0.021	0.060	0.073	0.202	0.998	0.826	0.674	0.525
CN600k→SCT	0.033	0.008	0.002	0.000	0.041	0.047	0.058	0.992	0.827	0.694	0.564
ROC→SCT	0.066	0.030	0.022	0.019	0.059	0.075	0.183	0.999	0.795	0.666	0.540
CN→SCT→Sen	0.072	0.035	0.027	0.024	**0.063**	0.077	0.201	0.998	0.854	0.721	0.588
CN→Sen→SCT	0.071	0.034	0.025	0.022	0.062	0.080	0.206	0.997	0.854	0.713	0.572
ROC→SCT→Sen	0.070	0.031	0.022	0.018	0.062	0.081	0.187	0.998	0.851	0.712	0.574
ROC→Sen→SCT	0.066	0.028	0.020	0.016	0.061	0.077	0.178	**0.999**	0.849	0.705	0.561
Ground Truth								0.979	0.897	0.794	0.691

Table 3. Overlap between generated and ground truth endings, and generated vs. gold ending diversity (*distinct*) scores for each model.

4.2. Models

While there are a few permutations, the following models are the key versions:

Base: The baseline GPT-2 model used is the smallest model, considered (Woolf, 2019) to be balanced in terms of performance and speed, with 768 dimensions, 12 layers, 12 attention heads, and 124M parameters.

Base to ConceptNet: The baseline model is post-trained with a single language modeling head on top of the transformer decoder, using the objective in Equation 1. Generated ConceptNet sentences (Eq. 3) are used as input, with tokens shifted to the right acting as labels. Output weights are tied to the input embeddings.

Base to ROCStories: The base model is post-trained as above, using the unlabeled ROCStories training data rather than ConceptNet sentences. The intention here is to compare incorporating commonsense assertions vs. simply training on more narratives.

Dataset	Train	Dev	Test
ConceptNet (LM only)	96,170	2,530	2,532
ROCStories (LM only)	93,360	2,393	2,408
ROCStories (SCT)	1,571	1,871	2,408
ROCStories (Sentiment)	1,571	1,871	2,408

Table 4. Dataset statistics for pure language modeling, Story Cloze Task (w/ auxiliary language modeling), and Sentiment Matching (w/ auxiliary language modeling). The ROCStories test set is used for generation prompts on any given model.

Base to Story Cloze Task: The base model is fine-tuned on labeled ROCStories data with the Story Cloze Task as the main task and the language modeling as auxiliary. Softmax with negative log-likelihood losses are averaged and combined as in Equation 2.

ConceptNet to Story Cloze Task: The base model is post-trained on the unlabeled ConceptNet data with language modeling as the objective, and then fine-tuned on the labeled ROCStories data with the Story Cloze Task as the main task and the language modeling as auxiliary.

ROCStories to Story Cloze Task: The base model is post-trained on the ROCStories data with language modeling as the objective, and then fine-tuned on held-out ROCStories data with the Story Cloze Task as the main task and the language modeling as auxiliary.

ConceptNet to Sentiment to Story Cloze Task: The base model is post-trained on the ConceptNet data with language modeling as the objective, and then fine-tuned on the VADER-annotated ROCStories data with the Sentiment Matching task as the main task and the language modeling as auxiliary, and again fine-tuned on the originally labeled ROCStories data with the Story Cloze Task as the main task and the language modeling as auxiliary.

ROCStories to Sentiment to Story Cloze Task: The base model is post-trained on the unlabeled ROCStories data with language modeling as the objective, and then fine-tuned on the VADER-annotated ROCStories data with the Sentiment Matching task as the main task and the language modeling as auxiliary, and then again fine-tuned on the originally labeled ROCStories data with the Story Cloze Task as the main task and the language modeling as auxiliary.

Slight variations are to reverse the SCT and sentiment tasks, and/or to combine or change the order of the ConceptNet and ROCStories post-training.

4.3. Settings

Hyperparameters are left the same as the original Radford et al. (2018) paper: Adam optimization with epsilon 1e-8, 3 epochs of training, joint loss weight $\lambda = 0.5$, weight decay 0.01, dropout 0.1, and batch sizes reduced to 16 to better fit a single standard GPU such as allotted for free by Google Colab[5]. The model uses learning rate

[5]https://colab.research.google.com

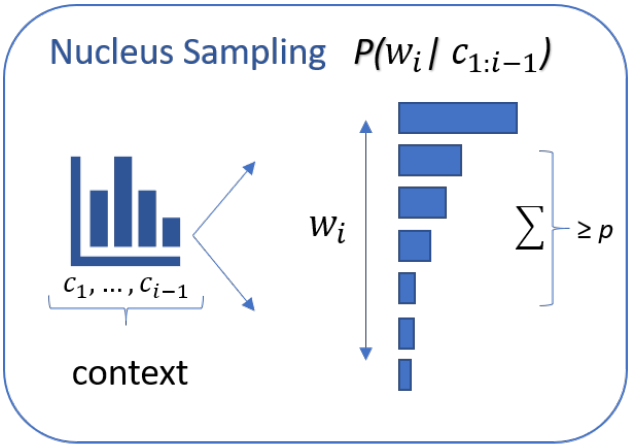

Figure 3. top-p nucleus sampling when predicting the next token conditioned on some context sequence. The smallest set whose cumulative probabilities surpass a threshold (e.g. 0.9) is selected as the more human-like truncated distribution.

Sentences	Number of words	Number of keywords
s_1	8.9	6.2
s_2	9.9	6.5
s_3	10.2	6.7
s_4	10.0	6.5
e_1	10.5	5.7
e_2	10.3	5.8

Table 5. Statistics from (Chen et al., 2019) showing the number of ConceptNet keywords in the ROCStories corpus (approximately 65%).

warm-up, a scheme for scaling the learning rate (starting from 6.25e-5) to minibatch size, which is intended to improve stability. For generation, top-k was set to 40, top-p to 0.9, temperature 0.7, and the repetition penalty to 1.2.

4.4. Evaluation

Models were evaluated by examining generated endings in relation to correct endings as well as generated endings in relation to story contexts. The most common automatic metrics appear to be overlap-based, although these are problematic for open-ended generation, as strict adherence to reference sentences is less meaningful to assess output quality. Nonetheless, these can be useful for prototyping, in lieu of human evaluations, therefore we used NLG–Eval[6] (Sharma et al., 2017) to evaluate hypotheses (generated endings) with respect to references (gold endings) using a combination of n-gram overlap and sentence embedding-based metrics, which

[6]https://github.com/Maluuba/nlg-eval

Model	Matches
Base	0.255399
Base→SCT	0.250415
ConceptNet→SCT	0.272425
CN600k→SCT	0.224252
CN→Sentiment	0.290698
ROC→SCT	0.298173
CN→SCT→Sentiment	0.274917
ROC→SCT→Sentiment	0.269518
CN→Sentiment→SCT	**0.319352**
ROC→Sentiment→SCT	0.304402
Gold	0.310631

Table 6. Percentage of endings matching their story context's average sentiment in terms of positive, negative, and neutral.

capture something of semantics. These metrics are not considered to correlate well with human judgments but can be useful for error analysis (Novikova et al., 2017).

In lieu of human evaluations, for this project we ran a varied assortment of automated metrics. Sharma et al. (2017) suggest that automated metrics such as BLEU correlate better with human judgment in settings where the task is more constrained (e.g., task-oriented dialogue responses such as scheduling an appointment). As this project focuses on generating single-sentence endings for brief stories, it's plausible that such metrics are somewhat more apt than if the generation task was more open-ended. The **overlap-based** metrics used by NLG–Eval include BLEU, METEOR, ROUGE, and CIDEr. **Embedding-based** metrics include Skip-Thought and Vector Extrema GloVe.

- **Perplexity** (PPL): The perplexity of respective versions of stories with correct vs. generated endings was evaluated by running the each version through the model and exponentiating the loss. Perplexity has been described as capturing diversity of generated text but not its quality (cf. human evaluations; Hashimoto et al. 2019; See et al. 2019).

- **Distinct** (Li et al., 2015) is a kind of type-token ratio measurement (unique n-grams divided by total sentence tokens).

- **Readability**: This uses spaCy's[7] implementation of the venerable ease of Kincaid et al. (1975) test to assess the complexity of generated endings based on ratios of lengths (words per sentence and syllables per word).

[7]https://spacy.io/

Model	SkipThoughtCS	EmbeddingAverageCS	VectorExtremaCS	GreedyMatchingScore
Base	0.625	*0.830*	0.363	0.672
Base→SCT	*0.672*	0.802	*0.380*	*0.672*
ConceptNet→SCT	0.658	0.771	0.362	0.659
ConceptNet600k→SCT	0.511	0.377	0.155	0.582
ROC→SCT	0.624	0.690	0.314	0.639
CN→SCT→Sent	0.666	0.795	0.370	0.666
CN→Sent→SCT	0.669	0.797	0.370	0.665
ROC→Sent→SCT	0.669	0.797	0.370	0.665
ROC→SCT→Sent	0.665	0.787	0.362	0.662
Gold	**0.690**	**0.857**	**0.435**	**0.722**

Table 7. Sentence-level embeddings' cosine similarities between story prompts and generated endings for each model. Generated models highest values in italics. Gold in bold.

- **Entity Coreference**: Following Roemmele et al. (2017)'s paper and code[8], we examined entity coherence between stories and generated endings using Stanford's CoreNLP[9]. Entity coherence is the coreference rate between generated sentences and their story prompts. That is to say, entity coherence measures the proportions of endings' entities that coreference entities from the context.

Model	**avg_ent**	**avg_coref**	**avg_res_rate**
Base	*3.330*	1.208	0.400
Base→SCT	2.151	1.199	0.583
CN→SCT	2.015	1.068	0.540
CN600k→SCT	1.850	0.827	0.410
CN→Sent→SCT	2.196	*1.215*	0.592
ROC→Sent→SCT	2.067	1.186	*0.605*
ROC→SCT	2.132	1.156	0.541
Gold	2.793	**1.817**	**0.645**

Table 8. Results for entity coreference between story prompts (averaged) and generated endings. Highest per generated model in italics, gold in bold.

4.5. Overlap-based metrics

- **BLEU** (Papineni et al., 2002) is an easy, popular and much criticized corpus-level metric which is oriented toward precision, the score weighted and penalized by brevity. Here it is evaluated on varying *n*-gram lengths, from unigrams to 4-grams.
- **ROUGE-L** (Lin, 2004) is a recall-oriented *F*-measure which evaluates the longest common subsequences, such that longer LCS indicates greater similarity.
- **METEOR** (Banerjee and Lavie, 2005) is a unigram matching-based metric which takes a weighted, penalized harmonic mean of precision and recall, where the weighting is much higher for recall than for precision.
- **CIDEr** (Vedantam et al., 2015) is based on the intuition of consensus between references for a given candidate and incorporates TF-IDF and averaged cosine similarity.

4.6. Embedding-based metrics

These metrics compute cosine similarities between sentence-level embeddings.

- **Vector Extrema** (Forgues et al., 2014) takes the most extreme (whether positive or negative) word vector values for each dimension to obtain sentence embeddings for generated and reference sentences. This has been described as tending to ignore common words in favor of informative words (Liu et al., 2017).
- **Skip-Thought** (Kiros et al., 2015) here relies on a pretrained RNN sentence embedding encoder for similarity evaluation.
- **EmbeddingAverageCS** simply averages word embeddings to create sentence embeddings.
- **GreedyMatching** is the average of greedy matches of embeddings based on word-level cosine similarities.

Intuitively, small perplexity scores putatively correspond to higher fluency, while *distinct* evaluates the lexical diversity of generated sentences. Entity coreference gives a sense of how well generated endings make use of characters in stories. Less complex (more readable) evaluations putatively correlate with higher naturalness

[8]https://github.com/roemmele/narrative-prediction/
[9]https://github.com/stanfordnlp/stanfordnlp

(Novikova et al., 2017). Embedding-based comparisons between story prompt and endings suggest lexical cohesion (Roemmele et al., 2017).

4.7. Analysis

As seen in Table 3, the baseline GPT-2 model scores highest for the *distinct* measures. Given its low scores for other metrics, this may be more of an indicator of incoherence in the endings, relative to the prompts and to what one might expect from the conclusion of a story, as opposed to a random sequence of tokens. The base to Story Cloze Task (Base→SCT) model scores higher for the BLEU-*n* metrics as well as ROUGE-L and CIDEr, suggesting that a small amount of fine-tuning on the smaller ROCStories datasets can quickly fit the model to the vocabulary of the stories.

For the embedding-based metrics (Table 7), the results are highly uniform, suggesting that the models learn to generate semantically similar sentences, which we might expect given the asymmetry of the story prompts and the constraints of the special start and delimiter tokens. It may be expected that the Vector Extrema results are lower than Skip-Thought and others if the former leans toward informative words rather than the more prosaic, functional terms that seem to populate the corpus. Gold endings are significantly higher across all measures, and taken as a whole, gives me the sense that any form of fine-tuning here contributes little to lexical cohesion in either direction.

Sentiment matches (Table 6)–the percentage of matches between the average sentiment, labeled as in Table 2) of story bodies and endings–shows that perhaps there is something of cohesion for sentiment learned by the sentiment matching task (which used distance scores rather than heuristic labels of valence), as the models trained first on sentiment or purely on sentiment scored near or better than the gold endings. However, the ROC→SCT model did well also, perhaps picking up on annotation artifacts in the ROCStories data. Interestingly, sentiment matching fine-tuning was unable to increase matching percentages when applied after the SCT fine-tuning. It may be that the proportion of labels which changed in the sentiment matching task as compared to the SCT labels was not high enough to shift the weights significantly.

Next to the gold ending, entity coreference rate (Table 8) is highest for the model first trained on the ROCStories corpus for pure language modeling, followed by multi-task sentiment matching and then the Story Cloze Task. A close second is the ConceptNet version of this model, and these results can be easily explained as resulting from adding more training epochs on the ROCStories corpus.

Our impression here is that this indicates the ConceptNet post-training made little difference, with initial ConceptNet language modeling weights perhaps overwritten during the fine-tuning stages on Sentiment and Story Cloze, such that each model's effects are primarily due to the multi-task fine-tuning on the labeled, small ROCStories datasets. Losses (Figures 4 and 5) suggest little difference between the base and ConceptNet models during SCT+LM fine-tuning, and this pattern continues for the losses for the Sentiment→SCT fine-tuning. Qualitatively, both proper names and pronouns, from first-person to gender, are frequently maintained, despite the subword tokenization having difficulties with less common names in the dataset, giving mangled alterations.

	Readability
Model	**Generated**
Base	91.920
Base→SCT	92.876
CN600k→SCT	**97.691**
CN→SCT→Sent	88.483
CN→Sent→SCT	88.246
ROC→SCT→Sent	85.348
ROC→Sent→SCT	86.448
Gold	82.170

Table 9. Flesch–Kincaid readability scores for endings per model.

While multiple choice (Story Cloze Task) losses are lower for the base→ConceptNet→SCT model than the base→SCT model while higher than the base→ROC→SCT model, we also know that there is substantive overlap in vocabulary (5) between ConceptNet triples and the ROCStories sentences, perhaps giving the base→ConceptNet model better initial token representations for the ROCStories-based SCT task.

In the same vein, perplexities (Table 10) are lowest for the model which begins with ROCStories post-training and is followed by sentiment matching and the Story Cloze task, followed by the ConceptNet version as a close second. A similar trend occurs with the reverse ordering of SCT followed by Sentiment Matching, although there appear to be some outliers in the single generation loop which caused a larger perplexity for the ROC→SCT→Sentiment model.

Readability scores (Table 9) are best for the 600k version of the ConceptNet to SCT model, edging out the base and the base GPT-2 to SCT models without fine-tuning, showing that generally there is some increase in complexity of generated endings with additional training, moving down toward the baseline ease of the correct ending.

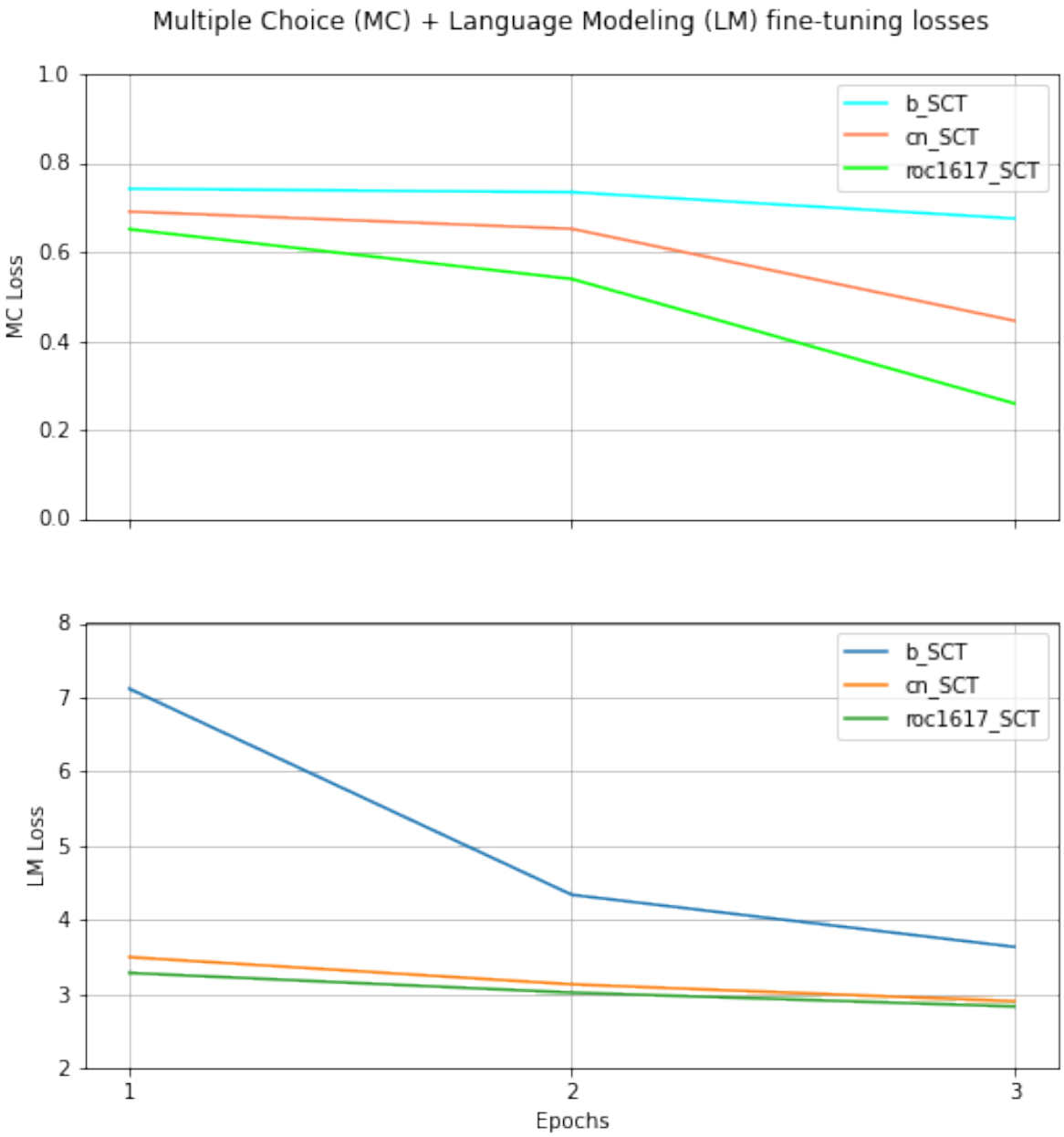

Figure 4. MTL fine-tuning losses for 3 epochs, as advocated by Radford et al. (2018): i.e., multiple choice (MC) and language modeling (LM) losses for the Base→ConceptNet model (pre-trained model after post-training (Eq. 1) on ConceptNet). Task losses are linearly combined during training (Eq. 2).

5. Conclusions and Future Work

This project attempted to fine-tune a pretrained language model, GPT-2, such that it would exhibit regularities associated with commonsense reasoning in the generation of story endings, where coherence in sentiment, entity coreference, and causal and temporal relations are maintained. A variety of training regimes were considered, intended to incorporate external knowledge through language modeling as well as multiple choice classification for both inference in selecting entailed endings and affect in selecting endings congruent in sentiment.

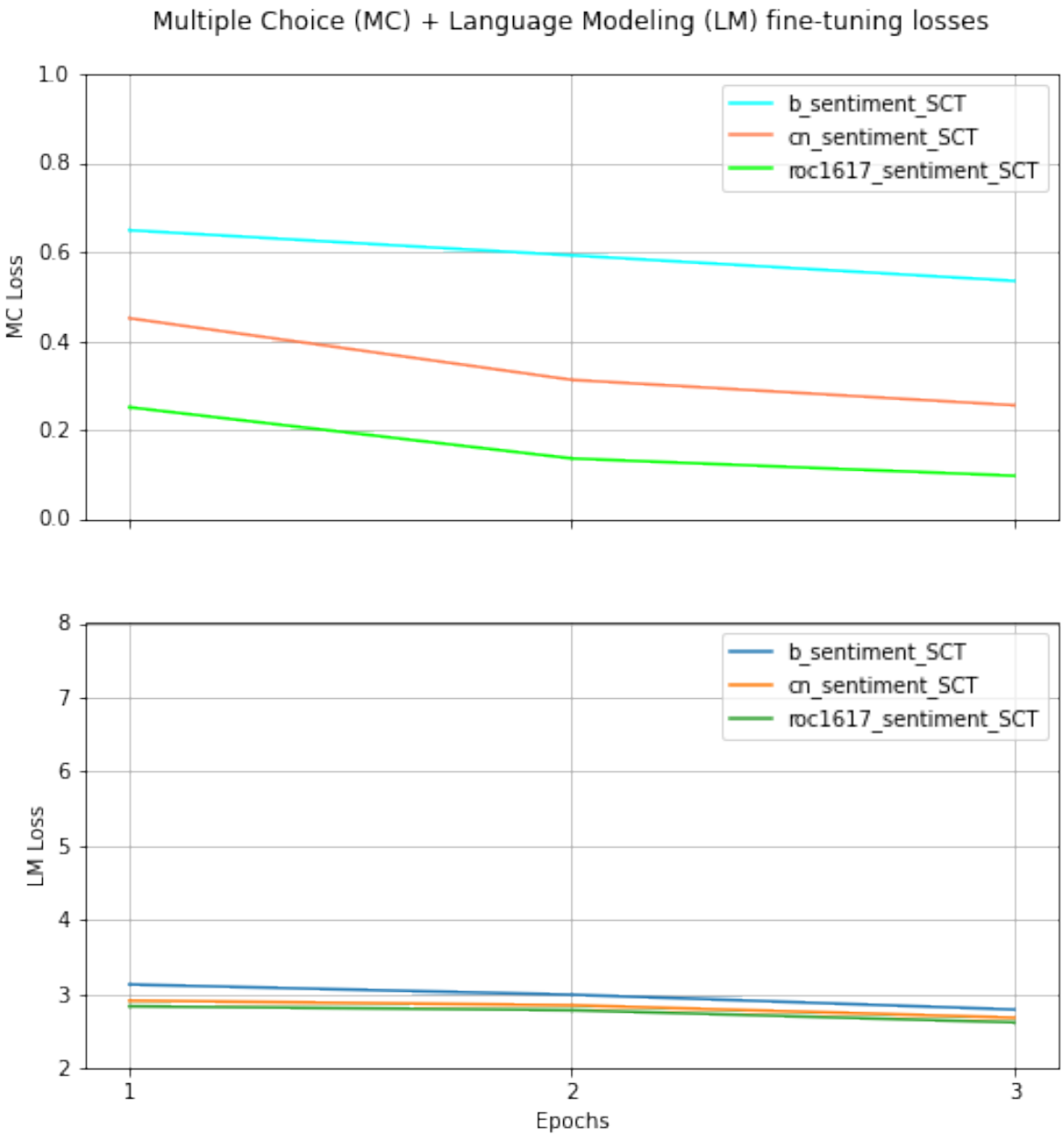

Figure 5. MTL fine-tuning (Sentiment→SCT) losses for 3 epochs, as advocated by Radford et al. (2018): i.e., multiple choice (MC) and language modeling (LM) losses for the Base→ConceptNet→Sentiment model (pre-trained model after post-training (Eq. 1) on ConceptNet and MTL fine-tuning on Sentiment Matching + LM). Task losses are linearly combined during training (Eq. 2).

Evaluation of generated endings with respect to story prompts as well as correct endings written by humans show relatively few differences between models. At this stage, it seems likely the Sentiment Matching task was not combined in a useful manner for multi-task fine-tuning. Using external knowledge data from 100k to 600k in size gave no significant benefits over merely adding more training with ROCStories data.

At present, further analysis in this area is difficult to pursue, as there are no clear methods or baselines for evaluating the contribution of semantic networks to open-ended text generation which point to common-

Model	Type	PPL
Base	Generated	242.382
	Golden	79.204
Base→SCT	Generated	242.548
	Golden	151.143
ConceptNet→SCT	Generated	179.693
	Golden	256.613
CN→SCT→Sentiment	Generated	41.136
	Golden	36.104
CN→Sentiment→SCT	Generated	34.760
	Golden	36.021
ROC→SCT→Sentiment	Generated	80.211
	Golden	33.666
ROC→Sentiment→SCT	Generated	**27.429**
	Golden	**30.025**
ROC→SCT	Generated	267.129
	Golden	34.069

Table 10. Perplexities for gold and generated endings for each model.

sense reasoning, as opposed to surface learning of characteristic artifacts in the data (Gururangan et al., 2018). Guan et al. (2020) suggest benefits may be evidenced by analyzing the perplexities the post-trained models assign to valid and invalid relations, where models which tend to give lower perplexity to valid relations–that is, continuations which contain relations and entity pairs found in the knowledge graph data–can be said to have learned causal relations.

However, we can expect the model to give lower perplexities to sequences which adhere to the ordering found in the training data, as the synthetic sentences in this data are template-based and thus the ordering of entities is unlikely to change, while each relation can be transformed into no more than a few predicates. Therefore, an alternate conclusion one might draw from such an evaluation is that the models are simply learning frequent orderings of certain triples which appear in the training data in regular, fixed ways. Instead, given the benefits of any increase in fine-tuning on ROCStories data, we might see this as a clue that the stories themselves are rich in thematic associations and provide a sort of textual grounding through self-contained events.

We might also consider that the base language model already contains the sort of knowledge of relations that the ConceptNet sentences are intended to represent and inject, through the enormous corpus the model was trained on. Work by Petroni et al. (2019) suggests that this indeed may be the case, although follow-up research (Kassner and Schütze 2019, Poerner et al. 2019) found that this knowledge is also rather superficial. This remains an open area of research, with approaches to knowledge infusion giving rise a slew of BERT variations.

Machine learning and deep learning approaches to narrative understanding and generation is a nascent subdomain with methodologies still being developed and tested, a frontier still being pioneered, and with a multitude of applications: from making sense of lists of facts as found in historical or biomedical records, producing edifying explanations from abstract processes or for educational materials, composing news articles–or detecting fraudulent news, to authoring narrative, creative works for entertainment as well as for psychological therapy. As language use is contextual and assistive tools depend on human feedback, feasible means of grounding and flexible controllability are two key areas to investigate.

Future work might better be directed toward exploring such grounding, examining the properties of narrative data in relation to generated text evaluations, and possibly exploring alternative means of generating data beyond current maximum likelihood or sampling-based approaches such as beam search and top-p. Controllability of features such as sentiment and adherence to relative cultural norms without the overhead of repeated finetuning of annotated data is another area to investigate, as well as the use of non-textual (e.g., visual) inputs in conjunction with textual inputs (both for training and generation), given the multimodal nature of language.

References

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. [§4.5.]

Lisa Feldman Barrett, Ralph Adolphs, Stacy Marsella, Aleix M Martinez, and Seth D Pollak. Emotional expressions reconsidered: challenges to inferring emotion from human facial movements. *Psychological Science in the Public Interest*, 20 (1):1–68, 2019. [§1.]

Emily M Bender and Alex Lascarides. Linguistic fundamentals for natural language processing ii: 100 essentials from semantics and pragmatics. *Synthesis Lectures on Human Language Technologies*, 12(3):1–268, 2019. [§1.]

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*, 2019. [§1.]

Samuel R Bowman, Christopher Potts, and Christopher D Manning. Recursive neural networks can learn logical semantics. *arXiv preprint arXiv:1406.1827*, 2014. [§1.]

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015. [§1.]

Jiaao Chen, Jianshu Chen, and Zhou Yu. Incorporating structured commonsense knowledge in story completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6244–6251, 2019. [§§2 and 5.]

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. Natural language inference with external knowledge. *arXiv preprint arXiv:1711.04289*, 2017. [§3.3.]

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008. [§1.]

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011. [§1.]

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer, 2005. [§1.]

Joe Davison, Joshua Feldman, and Alexander M Rush. Commonsense knowledge mining from pretrained models. 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 1173–1178, 2019. [§§1 and 3.1.]

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. [§1.]

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018. [§3.3.]

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*, 2014. [§2.]

Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. Bootstrapping dialog systems with word embeddings. In *NeurIPS, modern machine learning and natural language processing workshop*, volume 2, 2014. [§4.6.]

Philip Gage. A new algorithm for data compression. *C Users Journal*, 12(2):23–38, 1994. [§3.3.]

Pranav Goel and Anil Kumar Singh. IIT (BHU): System description for LSDSem'17 shared task. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 81–86, 2017. [§2.]

Jian Guan, Yansen Wang, and Minlie Huang. Story ending generation with incremental encoding and commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6473–6480, 2018. [§2.]

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. A knowledge-enhanced pretraining model for commonsense story generation. *arXiv preprint arXiv:2001.05139*, 2020. [§§2, 3.2, and 5.]

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018. [§5.]

Tatsunori B Hashimoto, Hugh Zhang, and Percy Liang. Unifying human and statistical evaluation for natural language generation. *arXiv preprint arXiv:1904.02792*, 2019. [§4.4.]

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019. [§3.3.]

Clayton J Hutto and Eric Gilbert. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*, 2014. [§§2 and 3.3.]

Nora Kassner and Hinrich Schütze. Negated lama: Birds cannot fly. *arXiv preprint arXiv:1911.03343*, 2019. [§5.]

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019. [§3.3.]

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. 1975. [§4.4.]

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015. [§4.6.]

Tassilo Klein and Moin Nabi. Attention is (not) all you need for commonsense reasoning. *arXiv preprint arXiv:1905.13497*, 2019. [§1.]

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015. [§4.4.]

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, page 74–81. Association for Computational Linguistics, Jul 2004. URL https://www.aclweb.org/anthology/W04-1013. [§4.5.]

Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004. [§1.]

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating Wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018. [§3.2.]

Bill MacCartney and Christopher D Manning. Natural logic and natural language inference. In *Computing meaning*, pages 129–147. Springer, 2014. [§1.]

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*, 2016. [§3.2.]

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for nlg. *arXiv preprint arXiv:1707.06875*, 2017. [§§4.4 and 4.6.]

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002. [§4.5.]

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? 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 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://www.aclweb.org/anthology/D19-1250. [§5.]

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. Bert is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised qa. *arXiv preprint arXiv:1911.03681*, 2019. [§5.]

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. [§§2, 3, 3.2, 4.3, 4, and 5.]

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. [§1.]

Andrew J Reagan, Lewis Mitchell, Dilan Kiley, Christopher M Danforth, and Peter Sheridan Dodds. The emotional arcs of stories are dominated by six basic shapes. *EPJ Data Science*, 5 (1):31, 2016. [§1.]

Melissa Roemmele, Andrew S Gordon, and Reid Swanson. Evaluating story generation systems using automated linguistic analyses. In *SIGKDD 2017 Workshop on Machine Learning for Creativity*, pages 13–17, 2017. [§§4.4 and 4.6.]

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. Do massively pretrained language models make better storytellers? *arXiv preprint arXiv:1909.10705*, 2019. [§§1 and 4.4.]

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015. [§3.3.]

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *arXiv preprint arXiv:1706.09799*, 2017. [§§4.4 and 4.4.]

Robyn Speer and Joanna Lowry-Duda. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. *arXiv preprint arXiv:1704.03560*, 2017. [§2.]

Shane Storks, Qiaozi Gao, and Joyce Y Chai. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*, 2019. [§1.]

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. [§4.5.]

Max Woolf. How to make custom ai-generated text with gpt-2, Sep 2019. URL https://minimaxir.com/2019/09/howto-gpt2/. [§4.2.]

Zhi-Xiu Ye, Qian Chen, Wen Wang, and Zhen-Hua Ling. Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. *arXiv preprint arXiv:1908.06725*, 2019. [§3.1.]

Albin Zehe, Martin Becker, Lena Hettinger, Andreas Hotho, Isabella Reger, and Fotis Jannidis. Prediction of happy endings in german novels based on sentiment information. In *3rd Workshop on Interactions between Data Mining and Natural Language Processing, Riva del Garda, Italy*, 2016. [§1.]

Yan Zhao, Lu Liu, Chunhua Liu, Ruoyao Yang, and Dong Yu. From plots to endings: A reinforced pointer generator for story ending generation. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 51–63. Springer, 2018. [§1.]

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629, 2018. [§2.]