Acquiring Narrative Understanding through Training with Short Stories

Probing Common-Sense Reasoning of Fine-tuned BERT Language Models

Eschbach, Johannes
j.eschbach@stud.uni-heidelberg.de

Eyubov, Kamal
hu229@stud.uni-heidelberg.de

Patzig, Leon
patzig@stud.uni-heidelberg.de

Sari, Raziye
sari@cl.uni-heidelberg.de

7th April 2021

1 Research Aim

In this paper, we intend to evaluate how well the state-of-the-art BERT model can be fine-tuned to understand common-sense relations in narratives. To perform well in such a task, a model needs to be able to understand casual and correlational relationships between events.

As training and test data, short stories seem the obvious choice. They are concise sequences of events rich in information. Furthermore, the events contained do typically display the named relationships between each other. Additionally, the short length of the stories allows us to easily manipulate the input for probing purposes.

Due to its contextualised understanding, BERT will have a cutting edge over its predecessors in the task at hand and should, in theory, be capable of identifying relevant events as well as the relationships between them required for basic common-sense reasoning. Whether training with short stories, however, is sufficient to encourage BERT to do so, will be analysed and discussed in this paper.

2 Method

With help of the ‘Story Cloze Test’ designed by Mostafazadeh et al. [MCH^{+}16] we will test the common sense reasoning of BERT and BERT-based fine-tuned Models. The Story Cloze Test consists of 5-sentence-stories – referred to as ‘ROCStories’ – whose final sentence is left out. The test then offers two final sentences to choose from. One of them fits the relative story, while the other violates the common sense established by it. We designed two concrete test scenarios for our models:

First, the model is charged with simply choosing the right final sentence out of both given options. Here, the model is not required to necessarily detect a common sense violation, as it just needs to state a preference. A human, however, would not even need an alternative ending to identify such a violation or inconsistency of the story. Hence, as a second test, we task the model with classifying whether a single given final sentence suits the story or not. We expect the models to perform worse in the latter test.

At first, we measure BERT's zero-shot performance on the tests. Second, we fine-tune BERT with a variation of training data sets and record the performances. Then, we use various experimental settings to probe the models, attempt to explain their performance and discover potential biases. Last, we will present our findings in regard to our initial research question.

3 Model

We use BERT as our NLU model.

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model built upon the Transformer architecture, consisting of a stack of multiple Transformer layers. In each of these layers a multi-head attention sub-layer as well as a feed-forward sub-layer are present.

What makes this state-of-the-art model greater than its predecessors is its ability to represent each given token in a input sentence as a contextualized word embedding. For that it takes into account three vectors: Apart from the pre-trained word embedding itself as 'Token Embeddings', a word's position in a sentence as 'Transformer Positional Embedding' plus the sentence it occurs in as 'Sentence Embedding'. BERT is pre-trained in two manners, both of which are used simultaneously in practice. One of them is Masked Language Modeling, where random tokens of the input sentence are [MASK]ed and the objective is to 'unmask' them as word outputs. The second method 'Next Sentence Prediction' can be seen as a Binary Classification problem such that BERT can learn whether a given next sentence is likely to follow the previous one. `[SEP]`arator tokens mark the end of a sentence and a binary output in the form of `[CLS]` is telling of BERT's decision whether sentence A follows sentence B.

BERT can easily be fine-tuned to quickly learn specific NLU tasks like Question Answering, Named Entity Recognition and the already known Classification task, also deployed for sentiment analysis. Small changes to the inner architecture of BERT e.g. a second output layer giving the answer to the Question-Answering task and supervised learning on data like the Stanford Question Answering Data set (SQuAD) are enough for fine-tuning BERT in order to perform great at special language tasks in a small amount of time.

With this deeply bidirectional and highly parameterized NLU model (BERT$_{\text{LARGE}}$ with 340 million parameters) we look forward to probe various common-sense reasoning tasks and analyze BERT's performances in zero-shot and fine-tuned settings.

4 Fine-Tuning data set

The ROCStories data set [MCH+16] will serve as fine-tuning data set.

ROCStories is a corpus of non-fictional short stories with everyday themes, specifically written by human crowd workers, aiming to cover diverse common-sense inter-event relations in a coherent narrative chain. To avoid inclusion of unnecessary information, every story consists of a title and five sentences, with a limit of 70 characters for each sentence. [MCH+16]

The ROCStories corpus covers 52 664 stories in the 'winter 2017' set, published later than the [MCH+16] paper, and can be requested at `https://cs.rochester.edu/nlp/rocstories/`.[1]

[1] Another version 'spring 2016' containing 45 495 stories is also available.

The authors assert high data quality by specifically designed tests. Meaningful temporal relations are demonstrated by other crowd workers, rearranging most of the sentences of shuffled stories back into their original order. [MCH$^+$16]

Apart from the ROCStories corpus, the authors have prepared validation and test sets specifically for the Story Cloze Test. These contain the first four sentences (no story title) and two options for the fifth (last) sentence; the validation set additionally contains information about the correct answer. Compared to the ROCStories corpus, these data sets are very small, as they only cover 1871 stories each. [SABM18]

5 Measuring Model Performances

We will employ BERT's next sentence prediction to predict the final sentences of the Story Cloze Test. The first four sentences of the respective ROCStory will be preceded by a `[CLS]` token, the last sentence separated by a `[SEP]` token. This sequence serves as input for the transformer-model. The output of the model is then fed through a classification layer, where probabilities for the binary classes of 'last sentence belongs to the sequence' and 'last sentence does not belong to the sequence' are calculated. In the easier test setup, the model determines the probability for each ending separately and then chooses the more likely one. This setup will from here on be referred to as 'Choice'. For the harder binary classification test setup, we will simply use the output provided by the model. From here on, this test setup will be referred to as 'Binary'.

5.1 Zero-Shot

As baseline for our fine-tuning, we measure the performance of the pre-trained BERT-for-next-sentence-prediction model.

Table 1: BERT: Story Cloze Test - Choice

Class	Precision	Recall	F1-Score
-	0.54	0.54	0.54
Accuracy		0.54	
Support		1871	

The two classes here are 'Ending 1' and 'Ending 2'. Since one ending is always correct and the other one is not, there is no point in distinguishing the two classes here. Furthermore, this test setup implies that precision, recall, F1-Score and accuracy are always equal.

The metrics show that BERT performs only slightly better than random-level. With a support of 3742 samples, however, a one-tailed z-Test between BERT's accuracy and random-level accuracy shows that the difference is statistically significant (0.0003 probability of error).

Table 2: BERT: Story Cloze Test - Binary

Class	Precision	Recall	F1-Score
True	0.54	0.19	0.28
False	0.51	0.84	0.63
Weighted average	0.52	0.51	0.45
Accuracy		0.51	
Support		3742	

In this test-setup, BERT has to identify common-sense violations without having a second ending for comparison. Without fine-tuning, BERT does not seem to be able to detect such violations. The accuracy of 0.51 surpasses random-level with hardly any statistical significance (z-Test: 0.19 probability of error). We assume that the low recall on true endings can be explained with the very concise format of the ROC-Stories, which BERT likely did not encounter during training. In authentic stories or reports, after all, it is hardly ever the case that every sentence introduces a new event.

In both test setups, the performance of BERT is below our expectations. With such a low baseline, however, we are certain that the model's performance can be greatly improved through fine-tuning.

5.2 Fine-Tuning

We decided to train BERT in a binary manner. The opposing idea of calculating loss based on the difference between the probabilities assigned to the true and false ending was quickly scrapped due to underwhelming results in preliminary experiments. After all, the binary test setup is the harder one, and whatever BERT learns through its loss backpropagation this way, should serve it as well in the easier choice test setup.

5.2.1 ROCStories Data Set

While the ROCStories data set has the perk of being substantial in size, it does, however, not provide any negative data points. To compensate for this, we employ a similar technique originally used for BERT's pre-training. Namely, we create false data points by assigning each story the ending of another story. This is not ideal, as it will likely just reinforce the recognition of topic-relatedness between story and ending, rather than improving the detection of common-sense violations. In the Story Cloze Test, then, the model will have to deal with two endings both topically continuing the story.

Due to the large size of the training data, we are bound to train the model with only one epoch.

Table 3: ROCStories: Story Cloze Test - Choice

Class	Precision	Recall	F1-Score
-	0.71	0.71	0.71
Accuracy		0.71	
Support		1871	

Given the lopsided training, the model achieves a surprisingly high accuracy on the choice test setup.

Table 4: ROC: Story Cloze Test - Binary

Class	Precision	Recall	F1-Score
True	0.52	0.99	0.68
False	0.88	0.08	0.14
Weighted average	0.70	0.53	0.41
Accuracy		0.53	
Support		3742	

The results of the second test setup could not be more different from the first. Recall for false endings is small, the recognition of common-sense violations therefore hardly existent. It becomes clear, that, in the first setup, the model was making decisions between 'true' and 'truer' rather than 'false' and 'true'. This was to be expected, as, during training, the model could not observe false data points containing common-sense violations, but only endings placed out of context. A closer inspection of the true negatives and false positives confirms that the model tends to only classify endings as false, when they seem out of context. Nonetheless, the capability of correctly ordering the endings by likelihood should not be understated. As with the other fine-tuned models, however, we will probe the model's performance and uncover potential biases. This model will from here on be referred to as 'ROC Model'.

The weight change graphs obtained by comparing the model parameter weights before and after the training is difficult to interpret, since change values are relatively similar over the layers (Appendix 1.1). However, the weight changes (and not the bias changes) seem to be slightly higher at the self-attention head layers of the deeper encoder units. It might be implied that in the higher more abstract levels, the model learns more about the relationship between the word representations than about their actual meaning, since self-attention heads mostly function to learn relationships between different words or representations and their role relative to other words or representations.

5.2.2 Story Cloze Test: Training Set

Mostafazadeh et al. provide a separated validation and testing set equal in size. We will make use of the validation set as a training set to provide our models with examples of common-sense violations. The Cloze Test validation set covers 1871 stories with a true and false ending each. Naturally, we expect fine-tuning with this data set to achieve the best metrics. The model was trained with ten epochs.

Table 5: CLOZE: Story Cloze Test - Choice

Class	Precision	Recall	F1-Score
-	0.91	0.91	0.91
Accuracy		0.91	
Support		1871	

The fine-tuned model achieves impressive metrics in the choice based test. While the ROC model's accuracy dropped to near random levels in the second test setup, this model maintains its performance with only a slight drop:

Table 6: CLOZE: Story Cloze Test - Binary

Class	Precision	Recall	F1-Score
True	0.83	0.86	0.84
False	0.85	0.82	0.83
Weighted average	0.84	0.84	0.84
Accuracy		0.84	
Support		3742	

With a precision of 0.85 and a recall of 0.82 on false endings, we have hopes that the model learned to recognise common-sense violations based on the false samples in the training data. However, given the small size of the validation set the model trained on, the results of both test setups should be viewed with caution. This model will from here on be referred to as 'CLOZE Model'.

Upon looking at the comparison of the encoder layers with the base model (the weight change graph), it becomes evident that the deeper layers (layers closer to the output) of the model have undergone a relatively larger change than the shallow layers (Appendix 1.2). This can be interpreted as the model learning complex semantic meanings and relationships from the training set.

5.2.3 Mixed Data Set

As we suspect the CLOZE Model to be biased, we decide to create a further dataset by expanding the Cloze Test validation set by 5000 ROCStories with correctly and wrongly assigned endings each. This way we hope to account for potential trigger words contained in the true-ending class. Like in the ROC Model training set, true endings will now also occur in the false-ending class whenever they are wrongly assigned. Furthermore, we hope that increasing the size of the training set also diminishes over-fitting. The amount of 5000 ROCStories is also not arbitrarily chosen, as the model displays a rapidly increasing drop-off in accuracy with 6000 or more ROCStories added.

Due to the increased size of the training set, we have to reduce training on five epochs.

Table 7: MIXED: Story Cloze Test - Choice

Class	Precision	Recall	F1-Score
-	0.92	0.92	0.92
Accuracy		0.92	
Support		1871	

In the easier test setup, this model slightly surpasses the CLOZE model in performance. This comes highly unexpected, as, given the composition of its training data, we anticipated the model to settle somewhere between the results of the CLOZE and ROC Model.

Table 8: MIXED: Story Cloze Test - Binary

Class	Precision	Recall	F1-Score
True	0.78	0.92	0.84
False	0.91	0.74	0.81
Weighted average	0.84	0.83	0.83
Accuracy		0.83	
Support		3742	

Here again, the weighted averages of the metrics and the accuracy reach levels near the CLOZE model. Noticable, however, is that precision shifts towards the false-ending class and recall towards the true-ending class. This behaviour has already been observed with the model trained only on ROCStories. However, given the training set's proportion of 5000 ROCStories to 1871 validation set stories, the drop in F1-Score for the false-endings (from 0.83 to 0.81) is proportionally small.

Overall, the model performs very well. Probing experiments will show whether this model fulfills our expectations of being more robust against adversarial attacks. This model will further be referred to as 'MIXED model'.

Despite a somewhat similar performance to the CLOZE model, this model has a different weight changes according to the weight change graph (Appendix 1.3). Despite having similarities with the ROC model at the weight change of the self-attention head layers (with both models having higher weight change values towards the output), the other layers seem to have higher change values neither towards the input nor towards the output but rather towards the middle. Thus, it can be implied that the model is not learning meanings either too specific for the next sentence prediction task or specific for the input words or tokens themselves.

6 Probing Experiments

6.1 Hypothesis-only testing

In hypothesis-only testing, the model is tasked to classify the endings without knowing the story. Naturally, it should not be able to predict the endings as the required events and relationships to do so are lacking. Nonetheless, the fine-tuned models will most likely perform above random level, as they are hardly ever unbiased. In this probing scenario, the difference between random-level and the model's performance can be interpreted as a metric for its bias.

Table 9: Hypothesis Only: Accuracy Drop

Model	Accuracy Drop
BERT - Choice	0.54 → 0.50
BERT - Binary	0.51 → 0.50
ROC - Choice	0.71 → 0.52
ROC - Binary	0.53 → 0.51
CLOZE - Choice	0.91 → 0.71
CLOZE - Binary	0.84 → 0.64
MIXED - Choice	0.92 → 0.58
MIXED - Binary	0.83 → 0.52
Support	Binary: 3742 Choice: 1871

The ‘Accuracy Drop’ column reads as: ‘Accuracy on unmanipulated test set’ → ‘Accuracy on hypothesis only test set’.

BERT achieves truly random level results in hypothesis only testing. As the full test-results show (Appendix 2.2.), this is because BERT simply considers every ending as false. This behaviour is indeed desired in such a test, as it shows that the model, with no context given, considers every sentence as a member of the same class. We will interpret any difference to BERT’s 0.5 accuracy as bias induced through fine-tuning. For instance, the CLOZE model shows high bias in both test scenarios. Most interesting, however, is the performance of the MIXED model, particularly in the binary test setup. Albeit achieving a CLOZE level accuracy in the normal story cloze test, hypothesis-only testing can not detect any more bias than there is in the ROC model. Furthermore, a closer inspection of its recall values shows that the MIXED model mimics the desired behaviour displayed by BERT and overwhelmingly considers the endings as false (Appendix 2.2.).

Two insights can be learned from this probing experiment. First, further investigation of the CLOZE model, as well as the validation set it was fine-tuned with, is needed to identify the causes for the high bias. In regard to our research question, the performance of the MIXED model provides us with evidence that models can be fine-tuned with short stories to identify the common-sense relations they contain.

6.2 Inducing Noise

Inducing noise into a data set is a common way to probe models for degenerate behaviour. In image processing, for instance, this can be done by adding a randomized pixel map to an image without affecting the human recognition of the image content. Such an operation is not as simple in language processing. However, we can use the opportunity and particularly challenge our models by including noise in a more motivated way.

In written and spoken language, we use discourse markers - often discourse connectives - to structure narratives. Therefore, a language model likely gains knowledge on events and their relations by looking at these discourse markers. For instance, the word ‘therefore’ establishes a causal relation between two events. These words, in turn, allow a model to identify such relations without needing to comprehend the semantic meaning of the two sentences. As our models are tasked with identifying common-sense relations through semantic understanding of the sentences involved, we will attempt to throw them off by placing discourse markers at random into the stories. Furthermore, for further noise, we induce temporal and locative adverbials, but not more than one each per story.

A long time ago Mary and Emma drove to the beach. In India, They decided to swim in the ocean. Consequently, Mary turned to talk to Emma. Meanwhile, Emma said to watch out for the waves.
1. The ocean was a calm as a bathtub.
2. A big wave knocked Mary down.

Just now When I was a kid I really wanted to play checkers. Therefore, I sat down with my grandpa and he taught me. Furthermore, At first, he was just teaching me but it became a special thing. Thereafter, As I got older, I continued to play checkers with him.
1. To this day I hate checkers.
2. I have very fond memories of checkers.

Sample 1: Noise Induced Stories

As the sample shows, the noise should hardly affect a human's recognition of the commonsense relation, but might just be enough to show degenerate behaviour in the models.

Table 10: Noise Induced: Accuracy Drop

Model	Accuracy Drop
BERT - Choice	0.54 → 0.54
BERT - Binary	0.51 → 0.51
ROC - Choice	0.71 → 0.70
ROC - Binary	0.53 → 0.53
CLOZE - Choice	0.91 → 0.90
CLOZE - Binary	0.84 → 0.84
MIXED - Choice	0.92 → 0.91
MIXED - Binary	0.83 → 0.83
Support	Binary: 3742 Choice: 1871

The models experience either no or only a small drop in accuracy. While the induced noise does not influence the final metrics, a closer look at the salience maps of binary testing shows that they do indeed cause the models to behave oddly (Appendix 3.1): While the noise-words themselves do not seem to have much importance for the classification, they do seem to trigger significant changes in the salience scores of other words. Ideally, however, the salience scores should not be affected at all, as the newly added words do not add any relevant information. Hence, we can conclude that the model indeed shows some degenerate behaviour, albeit with little effect on the final classification results.

6.3 Trigger Words

The experiments with trigger words aims to find out how much a presence of certain words (referred to as trigger words from now on) in a possible ending affects the prediction of a model. A trigger word has to have the following properties: It is frequent in the test set and removing it does not change the meaning of the sentence in any significant way. In order to find the trigger words, the following steps were carried out:

1. The ending sentences from the test set are analyzed and for each token, the token itself, the number of occurrences in negative endings and the number of occurrences in positive endings is added to a list. The tokens with less total occurrences than a certain number (for our experiments, we choose 10) are eliminated from the list.

2. For each element or token in the list, two new lists are created. The first list contains for each token, the token itself, the number of occurrences in positive endings, the pointwise mutual information (PMI) between the token and the positive ending class and the positive endings class likelihood for the token. The second list contains the same information but for the negative ending class. Both lists are sorted by the parameter corresponding to the PMI. This

step is only done for later analysis purposes and does not affect the choice of the trigger words.

3. Finally, the words are chosen from the tables of tokens and values calculated in the previous steps. This step is done manually, since it requires an actual understanding of the meaning of the words and their effect on sentences.

The first two steps produce lists with tokens like ‘saved’ with a PMI with the positive class of 0.60 and a positive class likelihood of 91% or ‘hates’ with a PMI with the negative class of 0.69 and a negative class likelihood of 100%. These words, however, are not chosen as trigger words, since removing them would make sentences either grammatically incorrect or change their meaning in a significant manner. After the third step the following tokens are obtained: ‘instead’, ‘ever’, ‘anymore’, ‘eventually’, ‘immediately’, ‘anyway’, ‘soon’, ‘later’, ‘now’, ‘finally’. A common feature among these tokens is that they are all adverbs.

After obtaining these words, three test sets are generated: StoryCloze set only with rows/examples where at least one of endings contains a removable trigger word, referred to as ‘Triggers Only’[2], a set referred to as ‘Triggers Removed’ obtained from ‘Triggers Only’ by removing the trigger words and a set referred to as ‘Triggers Synonymized’ obtained from ‘Triggers Only’ by synonymizing the trigger words (replacing trigger tokens with tokens of a similar meaning). Information about each trigger word, its removal and synonymization is described below.

- **instead**: PMI with negative class of 0.554997, negative class likelihood of 87.0968% This token only implies that something contrary to the context is happening. It can only be removed if the next token is not ‘of’. If it can be removed, it can also be synonymized with the token ‘alternatively’.
- **ever**: PMI with negative class of 0.518794, negative class likelihood of 84% This token serves to expand the past time window within a sentence. It can only be removed if the previous token is not ‘than’. If it can be removed, it can also be synonymized with the tokens ‘at any point’.
- **anymore**: PMI with negative class of 0.348307, negative class likelihood of 70.8333% This token implies continuation. It can be synonymized with the tokens ‘any further’.
- **too**: PMI with negative class of 0.0512933, negative class likelihood of 52.6316%. It can only be removed if it is at the end of the sentence. During its removal, the preceding comma is also removed. If it can be removed, it can be synonymized with the tokens ‘as well’.
- **eventually**: PMI with positive class of 0.693147, positive class likelihood of 100%. It can be synonymized with the token ‘ultimately’.
- **immediately**: PMI with both classes of 0, class probability of 50% for both classes (no correlation with any of the classes).
  This token implies the time frame after the last event of the context. It can be synonymized with the token ‘instantly’/
- **anyway**: PMI with the positive class of 0.336472, positive class likelihood of 70%. It can be synonymized with tokens ‘any ##how’ which together form the word ‘anyhow’. The word itself is not included as a token because of the tokenization used for this project.

[2] The need for this set arises from the fact that only a relatively small subset of the original Story Cloze test set contains endings with trigger words. Differences in the behavior of the model are clearer while comparing this particular subset and the following two sets.

- **soon**: PMI with the positive class of 0.336472, positive class likelihood of 70%. This is yet another token implying a time frame relative to the context. It can be removed if it is not preceded by the token ‘as’. It can be synonymized with the token ‘shortly’ in case if it can be removed.

- **later**: PMI with positive class of 0.390866, positive class likelihood of 73.913%. It can only be removed if there is no preceding measure of time or length such as ‘two years’ or ‘a kilometer’. If it can be removed it can also be synonymized with the token ‘afterwards’, and the following tokens ‘on’ or ‘on ,’ are removed.

- **now**: PMI with positive class of 0.516216, positive class likelihood of 83.7838%. This token can be removed if it is followed by ‘.’ or ‘,’ (in which case ‘,’ is removed as well). It can be synonymized with the token ‘currently’.

- **finally**: PMI with positive class of 0.575364, positive class likelihood of 88.8889%. This token can be synonymized with the tokens ‘at last’.

An example for the modifications:
With triggers:
`Finally, the sink was repaired.`
Triggers removed:
`the sink was repaired .`
Triggers synonymized:
`at last , the sink was repaired .`
Note the punctuation and how modified sentences are all in lower case. It is caused by tokenizing and reassembling the sentences. It has, however, no effect on the behavior of the model, since the sentences are tokenized again.

Table 11: Trigger Words: Accuracy Change

Model	Accuracy Change
BERT - Choice	0.56 → 0.59 → 0.57
BERT - Binary	0.51 → 0.53 → 0.52
ROC - Choice	0.74 → 0.74 → 0.74
ROC - Binary	0.56 → 0.55 → 0.56
CLOZE - Choice	0.93 → 0.92 → 0.93
CLOZE - Binary	0.83 → 0.80 → 0.83
MIXED - Choice	0.95 → 0.92 → 0.94
MIXED - Binary	0.83 → 0.84 → 0.83
Support	Binary: 348 Choice: 174

The ‘Accuracy Change’ column reads as: ‘Accuracy on “Triggers Only” set’ → ‘Accuracy on “Triggers Removed” set’ → ‘Accuracy on “Triggers Synonymized” set’.

The worst accuracy results are delivered by BERT, the default model. Its performance is almost on par with the model trained only on ROCStories train set when binary testing is done. Interestingly, no certain conclusions can be made based on the accuracy values in the table, since most changes are slight. A look at the saliency maps is therefore needed (Appendix 3.2). For the model trained only on ROCStories, the saliency mapping doesn’t seem to change significantly which makes sense only because the trigger words were mostly present within Story Cloze test data set, the data of which was never accessed during its training. Thus, little or almost no training was done related to those words.

Upon examining the models CLOZE and MIXED, however, a case with the trigger word ‘anyway’ is rather interesting. Both models make the wrong prediction, although the token ‘anyway’ is highlighted blue. This might imply that the models learn some information relating to the trigger words, but their prediction still doesn’t solely depend on them. Overall, it is difficult to interpret the saliency mapping either. On one hand, the trigger words themselves seem to be assigned saliency values relatively close to zero. On the other hand, removing or synonymizing the trigger words changes the saliency mapping over a whole example rather randomly. Then again, the change is not sufficient to alter the prediction of the model in most cases.

6.4 Negated Endings

To probe whether our models can handle choice inversion, we decide upon modifying the first 100 instances of the dataset with simple verb negation of the two possible endings to each story. By doing this, in most cases, the right ending becomes very unlikely to follow with respects to common-sense reasoning because of how on par they are with the rest of the story and a simple throwing in of *not* jettisons the likelihood of it being true completely. Like out of context noise sentences though, a lot of the times the counter false endings do not quite literally represent an opposite stance as in 'I don't love you' in opposition to 'I love you' in which case double negating the false ending would transform it into a fair option at hand. Therefore negating the false ending, does not yield a likelier option to end on leading up the story-telling.

Trudey wanted to write novels for a living. She wrote one through traditional publishing means. It barely made enough to cover the advance she had received. She wrote another through self-publishing avenues.
1. Trudey didn't hope self-publishing would be more profitable.
2. Trudey didn't call her sister and ask her to come to dinner.

Sample 2: Negated Endings

An example of how negating doesn't make the to be true ending 2. any likelier

Syntactically speaking, a fair amount of these sentences consist of more than just one main clause i.e. 'Magda loved cats' in the form of 'Magda thought that she loved cats', where *thought* of the introductory pronoun-verb combination calls for a second main clause. Same for combinations with verbs like *decide, hope, suggest* etc. Negating solely the first main clause sufficed. Another great deal of endings have compound verb phrases, often combined with a coordinating conjunction like *and* as shown above in ending 2. In this case negating only the first verb with *didn't call* was enough to invert the meaning of the whole sentence where accordingly the verb in the past tense *asked* became *ask*. Other than in cases of compound sentences where at least two main clauses exist independently. Here, negating both verbs was necessary to uphold the consistency such as: 'Magda didn't love dogs because they didn't love her either.' Contrary, infinitive verbs as adverbs in sentences: 'Magda fed her cat **to keep** her alive.' giving the answer to the question *"Why?"* raised in the introductory main clause, didn't need negating by virtue of them being subordinate clauses. Here again negating only the introductory main clause 'Magda didn't feed...' got the job done just fine. Naturally, instances of already by themselves negated statements appeared and here we double negated in the form of *never not* and even *not not*.

Table 12: Negated Endings: Accuracy Drop

Model	Accuracy Drop
BERT - Choice	0.55 → 0.49 (-11%)
BERT - Binary	0.49 → 0.51 (+4%)
ROC - Choice	0.69 → 0.40 (-42%)
ROC - Binary	0.55 → 0.45 (-18%)
CLOZE - Choice	0.94 → 0.42 (-14%)
CLOZE - Binary	0.85 → 0.47 (-45%)
MIXED - Choice	0.95 → 0.48 (-49%)
MIXED - Binary	0.87 → 0.46 (-47%)
Support	Binary: 200 Choice: 100

The percentage values in parentheses describe the relative change of accuracy for better comparability.

BERT's zero-shot outperforms all other trained models with this modification to the dataset. When producing endings with absences of events and sometimes even yielding two rather unlikely follow-ups, we see with BERT a suspected tendency to classify as false. This tendency is astoundingly reinforced when in binary setup, suggesting that without the given comparison of the now implausible option of the false ending, the model warily classifies the actually likelier noise sentence also as false. This recall value is the third lowest after ones corresponding to *Hypothesis-Only* and the to be followed *Super Hard Test*.

Table 13: Recall Change

Class	Choice	Binary	MODEL
True	0.40	0.09	BERT
False	0.36	0.03	ROC

When trained, ROC registers its all-time low values for both precision and recall for false endings. This shows that the model loses all its ability to recognize common-sense reasoning violations. One reason for this change from rather classifying warily as false in zero-shot to then the strong preference to classify as true when trained on ROCStories, might be that the model learns to predict true as soon as the same names and entities in the stories show up in the endings as well, regardless the negation. This acquired context-sensitivity when classifying could have been for instance triggered with the word *self-publishing* in the above sample ending 1.

6.5 Paraphrasing

In further experiments, we will use paraphrased versions of all phrases from the data set. This will introduce further noise and also reduce the frequency of trigger words.

Mallinson et al. describe a way to automatically generate high-quality paraphrasations by using neural machine translation, by pivoting through additional foreign languages. A paraphrasation of a multi-sentence text is therefore generated by translating the original text to a foreign pivot language, and translating the result back, where this process is repeated for several candidate pivot languages and the result with the highest evaluation score is used in the end. They show that their concept outperforms phrase-based methods, where often false paraphrasations are produced due to the loss of context. [MSL17]

While they use a specifically designed neural network, the use of a publicly available service providing neural machine translations will suffice for our purposes. Particularly, we will use *DeepL Translator*[3], which provides translations between 24 languages of acceptable quality free of charge.[4] In our approach, we simply use this third-party service to translate the stories through a pipeline of one or more pivot languages and back to English. We aim to find paraphrasations preferably different from the original stories. To achieve this, we perform the procedure described above using a single pivot language, for every language available. We then have multiple paraphrased versions of the original data set, and need to evaluate to which degree these versions differ from the original.

As a metric m for the evaluation of this degree, we use tokenised and stemmed versions

[3] `https://deepl.com/translator`

[4] We also considered to generate additional paraphrasations using *Google Translate* (`https://google.com/translate`) offering even more languages, but the service took much too long to translate enough data, i.e. the data set of 1870 Story Cloze tests could not be translated into one foreign language within 12 hours, where with DeepL, we had the translations and back-translations of multiple pivot languages within this time; so we decided that using DeepL must suffice.

s', s for each paraphrased phrase and the corresponding original (using the NLTK WordTokenizer and PorterStemmer [BKL09]), and compute the the minimum of the edit distance e and the cardinality of the symmetric difference of the sets S of stems:

$$m(s, s') = \min\left\{e(s, s'),\ \left|S(s) \triangle S(s')\right|\right\}$$

As documented in Appendix 4, *ja* (Japanese) leads to paraphrasations most different from the original, with an average score of 3.9, followed by *sl* (Slovenian) with 3.1 and *zh* (Chinese) with 3.0.

For the most promising languages[5], we ran the paraphrasation procedure again, iterating for each of them through the residual languages as the second pivot language in the pipeline. The resulting summarised scores are also shown in Appendix 4.

We can see that in every language pipeline, there are phrases which are not paraphrased at all (min = 0), but most of the phrases reach at least a score of 2 (median). The average score of *ja* is actually even higher than some of the dual-pivot pipelines (ranging from 3.2 for *fi, sv* to 4.6 for *hu, ja*) which is interesting, because all the other single-pivot pipelines reach an average score below the latter range.

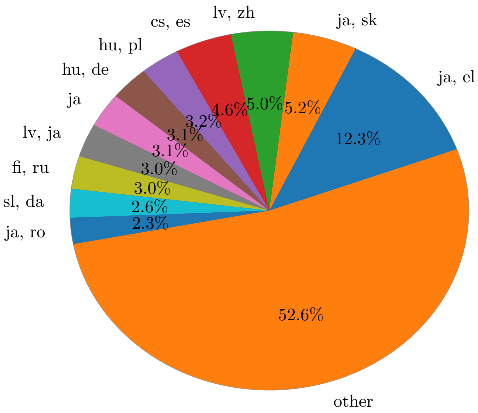

Figure 1: Fraction of the language pipelines used in the mixed paraphrased data set

On the one hand, we construct a data set made up of the maximum-score paraphrasation the for each phrase, so each phrase may be paraphrased through a different pipeline. When examining the paraphrasation data manually, it turns out that an important by-effect of the high scores for *ja* is especially caused by corrupted common-sense.

The two elderly ladies chatted as they ate lunch in the park. One of them brought out a basket and set the food out on the table. They talked and ate happily until they began to argue about something. One of them threw a piece of cake at the other, starting a food fight.
1. The police had to come to break up the argument.
2. They complemented each other on how tasty the cake was.

Sample 3: An original story to be paraphrased.

Two old ladies were chatting in the park, having lunch. One of the men took a basket and placed the food on the table. We talked and ate happily, but we argued about everything. One of them threw cake after cake and a food fight began.
1. The altercation had to be mediated by the police.
2. They thanked each other for the deliciousness of the cake.

Sample 4: The story, paraphrased using mixed language pipelines.

Interestingly, the pipelines used most often (see fig. 1) do not correspond directly to the pipelines with the highest average scores shown in Appendix 4.

[5] i.e. primarily *ja, sl, hu*; additionally *lv, et, fi, cs* but only partially (due to resource shortages)

When comparing the story from sample 3 with its mixed-pipeline paraphrasation shown in sample 4, we see that the phrases are indeed paraphrased, with changes in tense ('ladies chatted' → 'ladies were chatting'), choice of words ('elderly' → 'old') and clause position, which is desired behaviour. However, there are also incorrect changes of the genus ('One of them' [the ladies] → 'One of the men') and personal pronouns ('they' → 'we').

These mistakes seem to occur particularly for the languages *ja* and *zh*, which could be due to the fact that they are based on ideographic scripts other than the residual languages available, all using alphabetic scripts. On that account, we excluded these two languages from the further procedure.

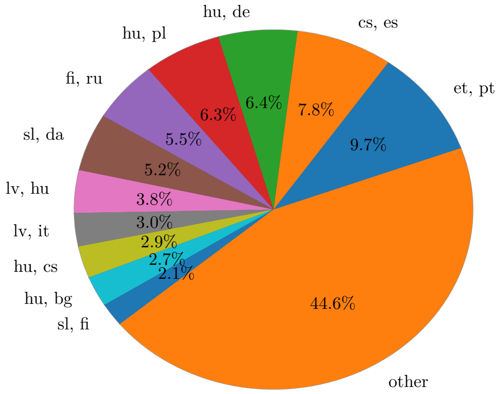

Figure 2: Fraction of the language pipelines used in the mixed paraphrased data set, when excluding *ja* and *zh*

> Two old ladies were chatting in the park, having lunch. One of them picked up a basket and put the food on the table. They were happy to talk and eat until they started discussing something. One of them throws a piece of cake to the other, which triggers a food fight.
> 1. The fight had to be stopped by the police.
> 2. They thanked each other for the deliciousness of the cake.

Sample 5: The story, paraphrased using mixed language pipelines, without *ja* or *zh* (Japanese or Chinese).

Having constructed the same data set, excluding language pipelines that involve *ja* or *zh*, we see that the problems described above have decreased, while the overall paraphrasation quality appears acceptable. The sample story from above is paraphrased in an entirely correct way (sample 5).

> Two elderly ladies were chatting during lunch in the park. One of them took a basket and put the food on the table. They were happy to talk and eat until they started arguing. One of them threw a piece of cake at the other and started a food fight.
> 1. The police had to break up the dispute.
> 2. They thanked each other for how delicious the cake was.

Sample 6: The story, paraphrased through the pivot languages *hu, sl* (Hungarian, Slovenian).

On the other hand, we used the entire data set of the language pipeline with the best average evaluation score, which is *hu, sl* in this case. The sample story above is also paraphrased differently though perfectly correct, as shown in sample 6.

However, after manual examination, there still remain some semantic mistakes in most of the paraphrased data sets, especially the problem with wrong pronouns.

Table 14: Paraphrasing: Evaluation Scores

	hu, sl	mixed
number of phrases	11226.0	11226.0
mean	3.8	6.3
std	2.7	2.8
min	0.0	0.0
25%	2.0	4.0
50%	4.0	6.0
75%	6.0	8.0
max	16.0	16.0

mixed refers to the mixed data set with *ja* and *zh* excluded

Table 15: Paraphrasing: Accuracy Drop

Model	source	mixed	*hu, sl*
BERT - Choice	0.54	0.50	0.53
BERT - Binary	0.51	0.50	0.51
ROC - Choice	0.71	0.64	0.67
ROC - Binary	0.53	0.53	0.54
CLOZE - Choice	0.91	0.85	0.87
CLOZE - Binary	0.84	0.77	0.81
MIXED - Choice	0.92	0.83	0.87
MIXED - Binary	0.83	0.75	0.78
Support	Binary: 3742 Choice: 1871		

mixed refers to the mixed data set with *ja* and *zh* excluded

As shown in table 15, the accuracies drop moderately for both the *hu, sl* and the mixed data sets, for all of the models except for the baseline BERT and ROC model with the binary test applied respectively. The drops on the mixed data set naturally go further than on the *hu, sl*, because the latter is less different from the original data set, as mentioned above.

These drops in accuracy might indicate the weakness of our models. Nevertheless, these results must be treated with caution, since the data sets are subject to occasional semantic mistakes.

6.6 Super Hard Test Set

After probing the models in various ways, we decided to create a super hard test set by applying to individual manipulations in a pipeline-like manner on the test set. As the negated test set is the only one requiring manual manipulation, it will serve as input for further automatized data manipulation. For paraphrasing, the stories are translated to Hungarian, then to Slovenian and then back to English. The pipeline looks as follows:

> Negate → Synonymise trigger words → Paraphrase → Induce noise

For comparison, the models' performance on non-manipulated data was measured on the same 99 stories the super hard test set comprises. Nonetheless, due to the small size of the test set, one should keep in mind that statistical error margins exist and the accuracy drop-off measured on the whole test set might differ slightly.

Table 16: Super Hard Testset: Accuracy Drop

Model	Accuracy Drop
BERT - Choice	0.55 → 0.47 (-15%)
BERT - Binary	0.49 → 0.48 (-2%)
ROC - Choice	0.69 → 0.43 (-38%)
ROC - Binary	0.55 → 0.56 (+2%)
CLOZE - Choice	0.94 → 0.55 (-41%)
CLOZE - Binary	0.85 → 0.55 (-35%)
MIXED - Choice	0.95 → 0.61 (-36%)
MIXED - Binary	0.87 → 0.54 (-38%)
Support	Binary: 198 Choice: 99

The percentage values in parentheses describe the relative change of accuracy for better comparability.

In the choice based test setup, BERT's accuracy drops by 15% when tested on the manipulated data. The fine-tuned models, however, experience much greater losses and therefore can be considered more biased than BERT. In binary testing, BERT barely loses any accuracy. While the other fine-tuned models again suffered from severe drops in accuracy, the ROC model even slightly improved. However, this value is deceptive. A closer inspection of the test metrics (Appendix 2.9) reveals that both BERT and ROC simply maintain their strong preference for one class, which by itself is a considerable bias. Further analysis shows that the ROC model simply continues only recognising those false data points whose endings are just as out of context after manipulation as they were before.

Due to BERT's bias for the false class in the binary test setup, we should consider the 15% accuracy-loss in choice based testing as true baseline for the other models. Even then, however, the experiment confirms that fine-tuning induced a significant amount of bias into the models.

7 Results

The fine-tuned models showed promising potential at first. However, probing experiments presented sobering results. Testing the models on hypothesis-only and negated stories as well as on the super hard test set revealed significant quantitatively measurable bias induced through fine-tuning. Even though the other experiments could not pin bias down as easily, a closer qualitative investigation then still detected unwanted or degenerate behaviour.

Although they are hard to interpret, a quick look at a randomly drawn sample of salience maps (binary testing mode) further reinforces this observation (Appendix 3.3.): We can observe spikes in tokens a human would not consider important for the sentence's sense. For instance, determiners, prepositions and conjunctions often get assigned extreme salience values. Also, the verb carrying out the action in the sentence often gets assigned very little importance.

Overall, the fine-tuned models all turned out to be flawed. While the CLOZE model achieved great test metrics, it was quickly exposed as also the most biased one, with a score of 0.71 accuracy in choice based hypothesis-only testing and a drop by 41% in the super hard test set. The ROC model achieved good results only in choice based testing. In this test setup, as especially the hypothesis-only testing showed, it is considerably less biased than the CLOZE model. Then again, in binary testing, it shows strong bias towards the 'true' class. It shows no ability to reliably detect common-sense violations, but rather just recognises out of context endings. If we were to rate the models, we would nominate the MIXED model as the one with best trade-off between test metrics and bias. However, even this model showed too high bias as well as too inconsistent salience maps to be considered a model that comprehends common-sense relations or violations

With the model evaluation at hand, we need to address the shortcomings in the training data employed by us. The ROCStories training set simply did not provide common-sense-violations. The Story Cloze test data comprising the test set and training set contained numerous trigger words and was likely just too small in size to be effectively used. Even though ROCStories and Story Cloze Test are created with high data quality controls in place, we believe that it would be better to create a data set by extracting multi-sentence sequences containing common sense relations from naturally occurring text, rather than employing crowd working.

Finally, we also need to consider whether BERT's next sentence prediction is the best choice for the research question at hand. The model is trained to make a prediction based on one preceding and one succeeding sentence each. However, in this paper, the model received four concatenated sentences where it would expect just one. We assume, that this might have contributed to the difficult intelligibility of the salience maps. In fact, BERT's NSP has been frequently criticized as ineffective and shallow due to the limited context provided by only one preceding sentence during training [AOR20]. Liu et al. [LOG^{+}19] particularly note that 'using individual sentences hurts performance on downstream tasks, which we hypothesize is because the model is not able to learn long-range dependencies'. We assume that this inability of the base-model also contributed to the fine-tuned models' incapability to reliably detect common-sense relations, which sometimes span multiple sentences after all.

8 Conclusion

We investigated whether BERT can be fine-tuned with short stories to show common-sense reasoning. We find that even with high quality crowd worked data, BERT can not be trained to reliably capture common-sense relations. We show that this is due to shortcomings in the data. Based on other research, we also believe BERT's next sentence prediction is not the best model to fine-tune for such purpose. We suggest to repeat this experiment with naturally occurring, rather than crowd-sourced data, and with a model specifically trained to capture dependencies of longer range than BERT's next sentence prediction does.

References

[AOR20] Aroca-Ouellette, Stephane ; Rudzicz, Frank: *On Losses for Modern Language Models*. 2020

[BKL09] Bird, Steven ; Klein, Ewan ; Loper, Edward: *Natural Language Processing with Python*, Juni 2009. `http://nltk.org/book/`

[LOG^{+}19] Liu, Yinhan ; Ott, Myle ; Goyal, Naman ; Du, Jingfei ; Joshi, Mandar ; Chen, Danqi ; Levy, Omer ; Lewis, Mike ; Zettlemoyer, Luke ; Stoyanov, Veselin: *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019

[MCH^{+}16] Mostafazadeh, Nasrin ; Chambers, Nathanael ; He, Xiaodong ; Parikh, Devi ; Batra, Dhruv ; Vanderwende, Lucy ; Kohli, Pushmeet ; Allen, James F.: A Corpus and Evaluation Framework for Deeper Understanding of Commonsense Stories. In: *CoRR* abs/1604.01696 (2016). `https://arxiv.org/abs/1604.01696`

[MSL17] Mallinson, Jonathan ; Sennrich, Rico ; Lapata, Mirella: Paraphrasing Revisited with Neural Machine Translation. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain : Association for Computational Linguistics, April 2017, 881–893

[SABM18] Sharma, Rishi ; Allen, James ; Bakhshandeh, Omid ; Mostafazadeh, Nasrin: Tackling the Story Ending Biases in The Story Cloze Test. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia : Association for Computational Linguistics, Juli 2018, 752–757