Comparing Data-Guided Techniques for Enhancing Negation Sensitivity in MLM-Based Laguage-Models

Philipp Koch
Institute for Statistics
Ludwig-Maximilians University
Munich
P.Koch@campus.lmu.de

Abstract

Despite the recent success with LLMs in NLP, unsolved challenges in the field of NLU remain. The problem of misunderstanding the concept of negation in natural languages has been investigated in previous years and remains unsolved. Different solutions to alleviate the problem have been introduced recently. This work examines different novel approaches using data-based strategies while keeping as much of the initial - promising - MLM objective as possible. We generate artificial data using WordNet to construct a setup of two sentences in which things from the real world are described using a standard description and a description using negation and antonyms for an adversarial sample. Additionally, we also create a negation-overrepresenting dataset by filtering RoBERTa's pre-training data. We conduct four experiments using this data and different masking strategies. The results indicate that using our training data with unsupervised MLM leads to slight overall improvements. We also observe stark improvements (51% to 62%) in accuracy but also strong fluctuating behavior. Our model also improves accuracy on WNLI (13 % to 41 %), although it drops on many other GLUE tasks.

1 Introduction

In recent years, large language models (LLMs) have proven to be a successful approach to solving many NLP-related problems. Especially the application of unsupervised learning on large amounts of data elevated the performance of LLMs (Devlin et al. (2019), Liu et al. (2019), Brown et al. (2020)). Specifically, the masked-language-model family has become a popular approach in natural language understanding (NLU) related tasks. Despite the unbroken success stories in many applications, some open problems remain regarding NLU. A particular issue in understanding is the concept of negation in natural languages, which may cause problems when it occurs in real-world applications. The problem has been the focus of research for quite some time (Ettinger (2020), Kassner and Schütze (2020), Hossain et al. (2020), Hossain et al. (2022), but it has not been sufficiently solved to this day. Different approaches exist where the problem can be alleviated using different objectives (Hosseini et al. (2021), Truong et al. (2022)). In this work, we investigate possible solutions approaching the problem mainly from the data side and investigate a supervised and unsupervised masking strategy to keep as much from the original - on many NLP-related tasks well-performing - MLM objective as possible.

2 Masked-Language-Modeling

Masked language modeling is a pre-training objective in NLP where a model is trained to predict a randomly masked word in a sentence based on the surrounding context. This technique was introduced by Devlin et al. (2019) based on Taylor (1953). The aim is to train the model to learn

contextual representations of words that can be fine-tuned for downstream NLP tasks. The loss function used in masked language modeling is typically cross-entropy loss. The model generates a probability distribution over the vocabulary for each masked word and is optimized upon this prediction and the original token.

BERT Bidirectional Encoder Representation Transformer (BERT) (Devlin et al.; 2019) is an encoder-only transformer model. The BERT model is trained on a large corpus of text data to learn contextual representations of words. This allows it to understand the meaning of a word in the context of the sentence it appears in. The model is bidirectional, meaning it can consider both the preceding and following words when generating a representation for a given word. BERT is trained using the MLM objective and next-sentence prediction.

RoBERTa In contrast to BERT, RoBERTa (Liu et al.; 2019) uses a different pre-training technique. The MLM objective and the share of masked words stay the same, but the training data is copied ten times, and the masks are applied randomly during training to allow the model to learn from different masking angles on the same sentence. Furthermore, the next-sentence prediction is dropped.

Both models achieved strong results and achieved state-of-the-art on various tasks at the time of their introduction.

3 Understanding of Negation in Transformers

The understanding capabilities of transformers have been extensively researched after the introduction of the BERT model. Ettinger (2020) conducted psycholinguistic experiments with it, including a task that examines the model's understanding of negation in natural language. The task consists of a fill-in-blank task in which the model is expected to predict a token in a negated context. The sentences included unnatural and constructed samples but also natural-sounding sentences. An example for the (unnatural) task is the sentence: *A robin is not a* `MASK`, where BERT-Large predicted: *robin, bird, penguin, man, fly*. However, for sentences with negation, it was expected to predict anything but *robin* or *bird*.

Although this task seems trivial for a human, BERT had severe problems solving it (BERT-Base: 38.9%, BERT-Large: 44.4% accuracy).

A similar approach to examine BERT's capabilities concerning negation was conducted by Kassner and Schütze (2020). The focus of this work is on factual knowledge of LLMs. Akin to Ettinger (2020), masked sentences were presented to the model for which the masked token was to be predicted. It was examined whether the model is capable of producing facts correctly. It was found that BERT (BERT-Base and BERT-Large) had a high overlap for correct and false answers when negation was added, which aligns with the findings of Ettinger (2020) of poor understanding of the concept of negation in BERT-models. However, it was also found that BERT can be improved in understanding negation when the model is fine-tuned in a supervised manner.

Further research was conducted regarding NLU corpora (Hossain et al.; 2022). It was found that negation does not appear in NLU corpora as commonly as in general-purpose English. Furthermore, negations are often unimportant to solving a task and do not even appear in some corpora. Models trained on these corpora yielded poorer results when they had to deal with tasks involving negation compared to tasks without negation. Additionally, natural language inference benchmarks were also examined regarding the importance of negation (Hossain et al.; 2020), where it was found that negation is often underrepresented and can also be ignored in some benchmarks to obtain correct results. A broader overview of the capabilities of LLMs was conducted by Lialin et al. (2022). This work includes the broader BERT family and decoder- and encoder-decoder-based models. The models were evaluated on a wide range of linguistic tasks using the oLMpics dataset (Talmor et al.; 2020), which also includes the task of antonym negation. It was found that RoBERTa obtained the highest value for the antonym negation task (RoBERTa-Large: 74.4%).

4 Related Work

Different approaches have already been introduced to alleviate the problem described in the previous section.

Hosseini et al. (2021) introduced an approach to training BERT on the meaning of negation by supervision. In this procedure, sentences are negated using a negation module, e.g., "A is a B." becomes "A is not a B.". Knowing that the word *B* is not desired at this position when also the first sentence is around, the model is trained to predict anything but *B*, using unlikelihood loss (Welleck et al.; 2019):

$$\mathcal{L}_{UL}(x_u) = -log(1 - p(x_B|x_{1:B-1}))$$

To further keep behavior from the original model, a distillation loss is used on unmodified samples, e.g., "A is a B." $\rightarrow$ "A is a B.":

$$\mathcal{L}_{KL}(x_l) = D_{KL}(p_{LM}||p)$$

In which p is the distribution of tokens from the original model, while p_LM is the distribution of the negation-aware model. The fine-tuned model showed improved results on negation-sensitive tasks than the original BERT.

Another approach to improving BERT focused on negation understanding was introduced by Truong et al. (2022). This work presented two pre-training techniques; both improved the model's performance on data from the medical domain. Negation in the medical domain often plays a crucial role in understanding a problem. For the first technique, the dataset is filtered, and only sentences with at least one negation are kept on which the model is subsequently trained using the model's original pre-training objective. The other technique enhances the masked-language-modeling objective by adding a new `[CUE]` token, which is only used to mask negation cues. Based on this novel masking strategy, the model is expected to learn a more in-depth relation of negation in natural language. The following sentence exemplifies this approach. The sentence *No serious complications such as hypertension, diabetes.* becomes *`[CUE]` serious complications such as `[MASK]`, diabetes.* Both approaches show improvement over the original BERT on some tasks.

5 Novel Techniques to Improve Understanding of Negation

This work aims to answer two questions regarding potential modifications in pre-training data to improve the model's capabilities with respect to the concept of negation.

Question 1: Does it suffice to train an LLM on data in which negation is overrepresented? Previous work (Hossain et al. (2022), Truong et al. (2022)) indicates that this approach might help improve the model's overall negation understanding, and it was found that negation is underrepresented in many corpora (Hossain et al.; 2022). To our knowledge, this approach had not been tested on the general pre-training data. Using this negation-aware data in general pre-training might improve the model.

Question 2: Can the model learn the concept of negation when specific (negation-aware) data is injected into the training data? The work of Kassner and Schütze (2020) and Hosseini et al. (2021) indicates a setup in which negation is passively explained through adversarial examples improves the model. Can this be potentially also used for general pre-training?

Due to the strong generalizability of the MLM objective, we deliberately try to be as close as possible to the initial MLM training and not modify the loss. The aim is to investigate if the current MLM pre-training objective can be enhanced using a different data-strategy.

6 Experiments

Three experiments will be conducted to answer the questions from the previous section. We train the model for three epochs and use the AdamW optimizer (Loshchilov and Hutter; 2019) for optimization in all experiments. We evaluate the model after ten thousand steps each and observe the behavior during training. Further hyperparameters are reported in the appendix.

Experiment 1: Training of BERT-Small on negation-aware (filtered) pre-training data of RoBERTa using the MLM objective (the selection of RoBERTa is intentional here since the highest score on the oLMpics antonym-negation task was achieved by this model). This experiment is similar to Truong

et al. (2022). However, we do not only train on domain-specific data but on the filtered pre-training data. Furthermore, we choose all sentences with at least one negation and their adjacent context of 2 neighboring sentences on each side.

Experiment 2 We train BERT-Small on adversarial data similar to Hosseini et al. (2021) and Kassner and Schütze (2020). Furthermore, we use instances from the filtered training data to avoid overfitting the artificial, adversarial training data. We apply supervised masking on the adversarial training data in which either the antonym or the word itself is masked. The model predicts the token based on the initial description and the negated description with the antonym. The setup is similar to Hosseini et al. (2021) but uses cross-entropy loss for optimizing the predictions to be closer to the original objective (because we have prior knowledge about the antonyms). At the same time, we keep random masking for filtered data.

Experiment 3 Instead of supervised masking on the adversarial data, we now mix filtered and adversarial data and use the MLM objective on the combined dataset in another attempt to test if masking words randomly is sufficient to improve the model's performance, approaching the problem entirely from the data.

The selection of BERT-Small is due to computational resources. We freeze the first three encoder layers of BERT-Small only to tune the upper layers. It was found that semantic knowledge is stored in the upper layers (Jawahar et al.; 2019), which we want to modify. For the implementation of the experiments, we use the transformers library (Wolf et al.; 2020) for this work.

7 Data

In this section, we will describe the data on which the experiments will be conducted. We use three datasets, of which two are solely for training and of which one is only for evaluation.

7.1 RoBERTa's Pre-Training Data

Due to the results in Lialin et al., that showed high accuracy of RoBERTa in antonym negation, we take the pre-training data of RoBERTa. RoBERTa was trained on five different datasets of which only four where available at the time of this work. The datasets used in this work are:

10% of available datasets (listed in the appendix) of the RoBERTa-Pre-Training dataset is searched for negations using spaCy's linguistic parser (Honnibal et al.; 2020). A short descriptive analysis of the results we found is provided in 4 (appendix).

For our purpose, we filter the dataset. All sentences that include a negation are collected. Since the context is assumed to be also required for the understanding of the negated phrase, we also add the previous two and the subsequent two sentences to the dataset. Using this technique, we can increase the share of sentences with negation to over 20%. An example of an instance from the filtered data can be seen in the appendix in Table 5. We will also refer to this data as *filtered data*.

7.2 Synthetic Data based on WordNet

Another approach is training the model using an adversarial data-setup, in which a sentence is used as an adversarial reference to teach the model the usage of negation. Hosseini et al. (2021) used unlikelihood loss for the adversarial reference sentence in which the word from the original sentence needs to be as far away from the antonym as possible in the latent space. Kassner and Schütze (2020) use a knowledge base to construct artificial sentences on which later BERT is trained in a supervised manner.

Our approach is similar, although no additional loss function will be used, and the knowledge base we are using is WordNet (Miller (1994), Fellbaum (1998), Princeton University (2010)). Both sentences describe something in the real world. The first sentence does this by simply stating the fact and the other sentence describes what the previous description is not by using negation and the antonym of the word in the initial sentence. Using this adversarial setup, we supply the model with an indirect description of how negation works in natural language. Table 1 shows an example of these sentences. Akin to the approach in Kassner and Schütze (2020), we build an artificial dataset by filling in templates using all available nouns, adjectives, and their respective antonyms from the WordNet

database. Verbs have not been used due to the difficulty of inflection in template-based systems. We used the NLTK (Bird et al.; 2009) and checklist (Ribeiro et al.; 2020) library to implement the dataset generator. Our dataset contains 755200 sentences and can be used in a supervised and unsupervised setup. An example of an instance of the supervised and unsupervised setup can be found in table 1. We will refer to this data as *WordNet adversarial data*.

	Data
Original	`This is a maximum. This isn't a minimum.`
Supervised	`This is a maximum. This isn't a [MASK]. [REF-BEG] minimum [REF-END]`
Unsupervised MLM	`This [MASK] a maximum. This isn't a minimum.`

Table 1: Example of an instance of WordNet adversarial data. (Special tokens in the supervised masking example are used to construct the labels during dataset instantiation.)

7.3 oLMpics Dataset for Evaluation

We use the oLMpics antonym negation task (Talmor et al.; 2020) to evaluate the model, similar to Lialin et al. (2022). This dataset consists of linguistic tasks in which the model has to predict a particular masked word. Since this prediction is a classification problem, accuracy is used as a metric. An example of an instance of the antonym negation task is: *It was `[MASK]` hot, it was really cold.*.

8 Results

The results of the trained BERT-Small model are reported in 2 and must be viewed in comparison to the baseline of 51 % accuracy on the oLMpics antonym negation task of the original BERT-Small model. For experiments 1 to 3, we refer to the appendix for the evaluation graphs.

	Experiment 1	Experiment 2	Experiment 3 (avg.)	Experiment 3+
Final Value	0.504	0.518	0.520	0.548
Best Value	**0.530**	**0.528**	**0.557**	**0.620**

Table 2: Results of the three experiments and the additional experiment 3+ based on experiment 3.

Experiment 1 Using only the filtered data to train BERT-Small with MLM objective did not improve the model significantly. At the beginning of the training, we observed slight fluctuation in which the performance rose to 53% accuracy at one evaluation step and dropped to the lowest point of 50.2% accuracy. After the 60th step, however, we observed stable behavior below the baseline of 51% accuracy. Eventually, the training converges to 50.04% accuracy.

Experiment 2 Training BERT-Small on WordNet adversarial data with supervised masking and on filtered data using unsupervised MLM also showed no improvement. The accuracy improved to 52.8% and dropped at the lowest point to 50% accuracy. Our final result was 51.8% accuracy, improving over the baseline of 51% by 0.8% percentage points. The training data used here was considerably smaller than in experiment 1.

Experiment 3 Training BERT entirely with an unsupervised MLM objective on both filtered and training data yielded a different behavior in contrast to the previous experiments. We trained the model three times using different seeds to validate this behavior. During the first steps in the first epoch, we observed stark fluctuations in accuracy for all three runs. The best value on average we obtained during the three runs was 55.7% accuracy, while the largest value in accuracy on an individual run was 57.2%. Although the values considerably increased, the behavior was unstable during this period (Epoch 1 in 1, appendix). In the latter two-thirds of the training, we observe a drop in accuracy while the fluctuations weaken and the procedure stabilizes. The average value of the three runs obtained at the end was 52% accuracy, which is a slight improvement over the baseline (51% accuracy). The training data used in this experiment included the same amount of instances as experiment 2.

Experiment 3+ Due to the improvement seen in Experiment 3, we tested another approach based on the pre-training strategy in RoBERTa. To further supply the model with even more data, we also

Figure 1: Evaluations during training of experiment 3+. Despite some stark improvements, the fluctuations also cause severe drops.

copied the WordNet adversarial data ten times, so the sentences can be masked and thus learned from different angles as introduced by Liu et al. (2019). Due to this procedure, the dataset size also increased substantially, limiting us to only train the model once. This experiment yielded an overall better result than the previous experiments. The highest value we obtained was 62% accuracy, while the lowest was 49.8% accuracy. We found even more fluctuations during training that continued into the second epoch than in the previous experiments, as seen in 1. The magnitude of the fluctuations were multiple percentage points. After the maximum value of 62% accuracy was reached, the model dropped in the next step to 54%. This unstable behavior stopped during the last epoch when the accuracy became more stable at a higher value than the baseline. Finally, the training finishes with an accuracy of 54.8%, which is an improvement of 3.8 percentage points over the baseline.

Throughout all experiments, we could not observe a clear upwards trend in accuracy (but a downward trend for experiment 1). The first two experiments did not deviate by more than 0.8% points from the baseline eventually. In comparison, the last two experiments partially caused an increase of multiple percentage points. However, both experiments showed stark unstable behavior at the beginning of the training, where the highest values were observed. Eventually, both approaches (exp. 3 and exp 3+) led to a slight improvement after training.

9 Evaluation on GLUE

	Metric	Original	Ours
Microsoft Research Paraphrase Corpus (MRPC)	Accuracy	**0.79**	0.66
	F1	**0.86**	0.76
The Corpus of Linguistic Acceptability (COLA)	Matthews Corr. Coeff.	**0.36**	0.12
Winograd Natural Language Inference (WNLI)	Accuracy	0.13	**0.41**
Question Natural Language Inference (QNLI)	Accuracy	**0.86**	0.57
Quora Question Pairs (QQP)	Accuracy	**0.90**	0.82
	F1	**0.86**	0.77
Recognizing Textual Entailment (RTE)	Accuracy	**0.65**	0.51
The Stanford Sentiment Treebank (SST)	Accuracy	**0.88**	0.79

Table 3: Results for the evaluation on selected GLUE tasks.

To verify whether the performance of the model is still kept after applying experiment 3+, we evaluate the final model on selected GLUE tasks (Wang et al.; 2018) [1]. Keeping the model competitive failed for most tasks, where our model dropped compared to the original BERT-small model. However, for the Winograd NLI Challenge (Levesque et al.; 2011) we observe a stark improvement by more than threefold. The results are reported in table 3 and the hyperparameters for fine-tuning are reported in 10 (appendix).

10 Discussion and Limitations

The validity of this work remains limited due to computational limitations that constrained the use of costly experiments and validations. Only experiment 3 was trained using three different seeds, where similar behavior between the training runs was observed. However, this was not possible for other experiments, which were too costly. Due to this limitation, it was also not feasible to explore the limits of the current state-of-the-art model (RoBERTa-Large), which might have behaved differently. Furthermore, the usage of large datasets was thus also not possible, and thus only 10% of the pre-training data of RoBERTa was used for experiment 1. Another limitation following this problem is the absence of strong augmentation models for WordNet adversarial data. While it was possible to generate data using nouns and adjectives by filling in templates, this did not work for verbs, for which inflection is necessary.

11 Conclusion

Although some improvements have been seen, the results remain still close to guessing. The final values of all experiments remain close to the initial baseline and at 50% accuracy. Furthermore, the fluctuating behavior during training weakens the power of the more significant improvements during training, especially shown by the sharp drop after achieving 62% accuracy down to 54%. Returning to the questions from section 5, we can not conclude that training solely on data in which more negation appears alleviates the problem (**question 1**). However, for **question 2**, we can see a slight improvement when training a model on the adversarial data, although the overall performance on general benchmarks (selected GLUE-tasks) drops. The slight improvement as soon as WordNet adversarial data is introduced aligns with the results of previous work (Kassner and Schütze (2020), Hosseini et al. (2021)), which also shows improvement using this setting. Similar to Truong et al. (2022), our approach of using filtered data from original pre-training and artificial adversarial data might be helpful to warm up models before fine-tuning them on negation-intense tasks as in the medical domain. Further research in this direction is highly appreciated. The proposed technique of training an MLM-based model on filtered and adversarial data significantly improved the model's capability in solving the Winograd NLI challenge. Our model achieves a more than threefold increase in accuracy on WNLI compared to the original BERT-Small, implying that the proposed technique might also be helpful in other tasks besides improving the model's negation understanding. The computational constraint problem also limits this work's validity. Thus, we recommend replicating the experiments using current state-of-the-art models and augmentation for WordNet adversarial data to also use verbs in the dataset.

References

Bentivogli, L., Clark, P., Dagan, I. and Giampiccolo, D. (2009). The fifth pascal recognizing textual entailment challenge.

Bird, S., Klein, E. and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*, " O'Reilly Media, Inc.".

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray,

[1] MRPC (Dolan and Brockett; 2005), COLA (Warstadt et al.; 2018), WNLI (Levesque et al.; 2011), QNLI (Rajpurkar et al.; 2016), QQP (Shankar et al.; 2016), RTE (Dagan et al. (2006), Haim et al. (2006), Giampiccolo et al. (2007), Bentivogli et al. (2009)), SST (Socher et al.; 2013)

S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D. (2020). Language models are few-shot learners.

Dagan, I., Glickman, O. and Magnini, B. (2006). The pascal recognising textual entailment challenge, *in* J. Quiñonero-Candela, I. Dagan, B. Magnini and F. d'Alché Buc (eds), *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 177–190.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186.
URL: *https://aclanthology.org/N19-1423*

Dolan, B. and Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases, *Third International Workshop on Paraphrasing (IWP2005)*, Asia Federation of Natural Language Processing.
URL: *https://www.microsoft.com/en-us/research/publication/automatically-constructing-a-corpus-of-sentential-paraphrases/*

Ettinger, A. (2020). What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models, *Transactions of the Association for Computational Linguistics* **8**: 34–48.
URL: *https://aclanthology.org/2020.tacl-1.3*

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*, Bradford Books.
URL: *https://mitpress.mit.edu/9780262561167/*

Foundation, W. (2022). Wikimedia downloads.
URL: *https://dumps.wikimedia.org*

Giampiccolo, D., Magnini, B., Dagan, I. and Dolan, B. (2007). The third PASCAL recognizing textual entailment challenge, *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Association for Computational Linguistics, Prague, pp. 1–9.
URL: *https://aclanthology.org/W07-1401*

Gokasla, A., Coohen, V., Pavlick, E. and Tellex, S. (2019). Openwebtext corpus, `http://Skylion007.github.io/OpenWebTextCorpus`.

Haim, R. B., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B. and Szpektor, I. (2006). The second pascal recognising textual entailment challenge, *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Vol. 7.

Hamborg, F., Meuschke, N., Breitinger, C. and Gipp, B. (2017). news-please: A generic news crawler and extractor, *Proceedings of the 15th International Symposium of Information Science*, pp. 218–223.

Honnibal, M., Montani, I., Van Landeghem, S. and Boyd, A. (2020). spacy: Industrial-strength natural language processing in python.
URL: *https://spacy.io/*

Hossain, M. M., Chinnappa, D. and Blanco, E. (2022). An analysis of negation in natural language understanding corpora, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Association for Computational Linguistics, Dublin, Ireland, pp. 716–723.
URL: *https://aclanthology.org/2022.acl-short.81*

Hossain, M. M., Kovatchev, V., Dutta, P., Kao, T., Wei, E. and Blanco, E. (2020). An analysis of natural language inference benchmarks through the lens of negation, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Online, pp. 9106–9118.
URL: *https://aclanthology.org/2020.emnlp-main.732*

Hosseini, A., Reddy, S., Bahdanau, D., Hjelm, R. D., Sordoni, A. and Courville, A. (2021). Understanding by understanding not: Modeling negation in language models, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Online, pp. 1301–1312.
URL: *https://aclanthology.org/2021.naacl-main.102*

Jawahar, G., Sagot, B. and Seddah, D. (2019). What does BERT learn about the structure of language?, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, pp. 3651–3657.
URL: *https://aclanthology.org/P19-1356*

Kassner, N. and Schütze, H. (2020). Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, pp. 7811–7818.
URL: *https://aclanthology.org/2020.acl-main.698*

Levesque, H. J., Davis, E. and Morgenstern, L. (2011). The winograd schema challenge., *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning* .

Lialin, V., Zhao, K., Shivagunde, N. and Rumshisky, A. (2022). Life after BERT: What do other muppets understand about language?, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Dublin, Ireland, pp. 3180–3193.
URL: *https://aclanthology.org/2022.acl-long.227*

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization.

Miller, G. A. (1994). WordNet: A lexical database for English, *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.
URL: *https://aclanthology.org/H94-1111*

Princeton University (2010). About wordnet.
URL: *https://wordnet.princeton.edu/*

Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Austin, Texas, pp. 2383–2392.
URL: *https://aclanthology.org/D16-1264*

Ribeiro, M. T., Wu, T., Guestrin, C. and Singh, S. (2020). Beyond accuracy: Behavioral testing of NLP models with CheckList, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, pp. 4902–4912.
URL: *https://aclanthology.org/2020.acl-main.442*

Shankar, I., Nikhil, D. and Kornl, C. (2016). First quora dataset release: Question pairs.
URL: *https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs.*

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Seattle, Washington, USA, pp. 1631–1642.
URL: *https://aclanthology.org/D13-1170*

Talmor, A., Elazar, Y., Goldberg, Y. and Berant, J. (2020). oLMpics-on what language model pre-training captures, *Transactions of the Association for Computational Linguistics* **8**: 743–758.
URL: *https://aclanthology.org/2020.tacl-1.48*

Taylor, W. L. (1953). "cloze procedure": A new tool for measuring readability, *Journalism Quarterly* **30**(4): 415–433.
URL: *https://doi.org/10.1177/107769905303000401*

Truong, T. H., Baldwin, T., Cohn, T. and Verspoor, K. (2022). Improving negation detection with negation-focused pre-training.
URL: *https://arxiv.org/abs/2205.04012*

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Association for Computational Linguistics, Brussels, Belgium, pp. 353–355.
URL: *https://aclanthology.org/W18-5446*

Warstadt, A., Singh, A. and Bowman, S. R. (2018). Neural network acceptability judgments, *arXiv preprint arXiv:1805.12471* .

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K. and Weston, J. (2019). Neural text generation with unlikelihood training.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q. and Rush, A. (2020). Transformers: State-of-the-art natural language processing, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Online, pp. 38–45.
URL: *https://aclanthology.org/2020.emnlp-demos.6*

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A. and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, *The IEEE International Conference on Computer Vision (ICCV)*.

A Appendix

A.1 Filtered Data

A.1.1 Available RoBERTa Pre-Training Datasets

- BookCorpus (Zhu et al.; 2015)
- English Wikipedia (Version 2022/03/01, Foundation (2022)
- CC-News (Hamborg et al.; 2017)
- OpenWebText (Gokasla et al.; 2019)

A.1.2 Descriptive Analysis of Negation in RoBERTa Pre-Training-Data

Dataset	Sent. w Negation (in %)
Wikipedia	5
BookCorpus	17
CC-News	12
OpenWebText	16
Total	11

Table 4: Percentage of negations in datasets used in RoBERTa pre-training

The amount of sentences with negations aligns with the findings of Hossain et al. (2022) as there is also a misrepresentation compared to general-purpose English in which negation occurs more often (22.6%–29.9%, Hossain et al. (2022)).

A.1.3 Example

Includes Negation ?	Sentence
w/o negation	emma rolled her eyes .
w/o negation	“ i ’m very satisfied with both my choices , megan .
w negation	you do n’t have to worry . ”
w/o negation	“ so who is the godfather again ?
w negation	he ’s not part of the family . ”

Table 5: Example of an instance of the filtered dataset from RoBERTa pre-training data.

A.2 Hyperparameters - Experiment 1

Parameter	Value
Model Identifier (transformers library)	`prajjwal1/bert-small`
Optimizer	AdamW
Learning Rate	5e-5
Epochs	3
Batch Size	16
Blocksize	128
Data Collator for Training	`DataCollatorForLanguageModeling` (transformer library)
Test Ds. Proportion	5
Steps (Evaluation)	10000
seed	42
Layers Frozen	0, 1, 2
Frozen Layers	bert.encoder.layer.{0,1,2,}.attention.self.query.weight bert.encoder.layer.{0,1,2,}.attention.self.query.bias bert.encoder.layer.{0,1,2,}.attention.self.key.weight bert.encoder.layer.{0,1,2,}.attention.self.key.bias bert.encoder.layer.{0,1,2,}.attention.self.value.weight bert.encoder.layer.{0,1,2,}.attention.self.value.bias bert.encoder.layer.{0,1,2,}.attention.output.dense.weight bert.encoder.layer.{0,1,2,}.attention.output.dense.bias bert.encoder.layer.{0,1,2,}.attention.output.LayerNorm.weight bert.encoder.layer.{0,1,2,}.attention.output.LayerNorm.bias bert.encoder.layer.{0,1,2,}.intermediate.dense.weight bert.encoder.layer.{0,1,2,}.intermediate.dense.bias bert.encoder.layer.{0,1,2,}.output.dense.weight bert.encoder.layer.{0,1,2,}.output.dense.bias bert.encoder.layer.{0,1,2,}.output.LayerNorm.weight bert.encoder.layer.{0,1,2,}.output.LayerNorm.bias

Table 6: Hyperparameters used in Experiment 1. All values that are not reported, are default values from the transformer library.

A.3 Evaluation Graph - Experiment 1

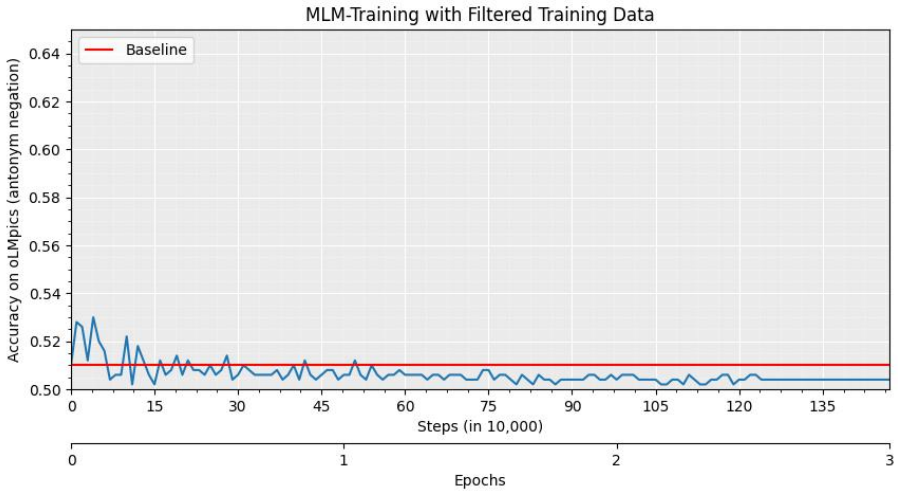

Figure 2: Experiment 1: Training BERT-Small on the filtered dataset.

A.4 Hyperparameters - Experiment 2

Parameter	Value
Model Identifier (transformers library)	`prajjwal1/bert-small`
Optimizer	AdamW
Learning Rate	5e-5
Epochs	3
Batch Size	24
Data Collator for Training	`DataCollatorForTokenClassification` (transformer library)
Test Ds. Proportion	5
Steps (Evaluation)	10000
seed	42
Layers Frozen	0, 1, 2
Frozen Layers	bert.encoder.layer.{0,1,2,}.attention.self.query.weight bert.encoder.layer.{0,1,2,}.attention.self.query.bias bert.encoder.layer.{0,1,2,}.attention.self.key.weight bert.encoder.layer.{0,1,2,}.attention.self.key.bias bert.encoder.layer.{0,1,2,}.attention.self.value.weight bert.encoder.layer.{0,1,2,}.attention.self.value.bias bert.encoder.layer.{0,1,2,}.attention.output.dense.weight bert.encoder.layer.{0,1,2,}.attention.output.dense.bias bert.encoder.layer.{0,1,2,}.attention.output.LayerNorm.weight bert.encoder.layer.{0,1,2,}.attention.output.LayerNorm.bias bert.encoder.layer.{0,1,2,}.intermediate.dense.weight bert.encoder.layer.{0,1,2,}.intermediate.dense.bias bert.encoder.layer.{0,1,2,}.output.dense.weight bert.encoder.layer.{0,1,2,}.output.dense.bias bert.encoder.layer.{0,1,2,}.output.LayerNorm.weight bert.encoder.layer.{0,1,2,}.output.LayerNorm.bias

Table 7: Hyperparameters used in Experiment 2. All values that are not reported, are default values from the transformer library.

A.5 Evaluation Graph - Experiment 2

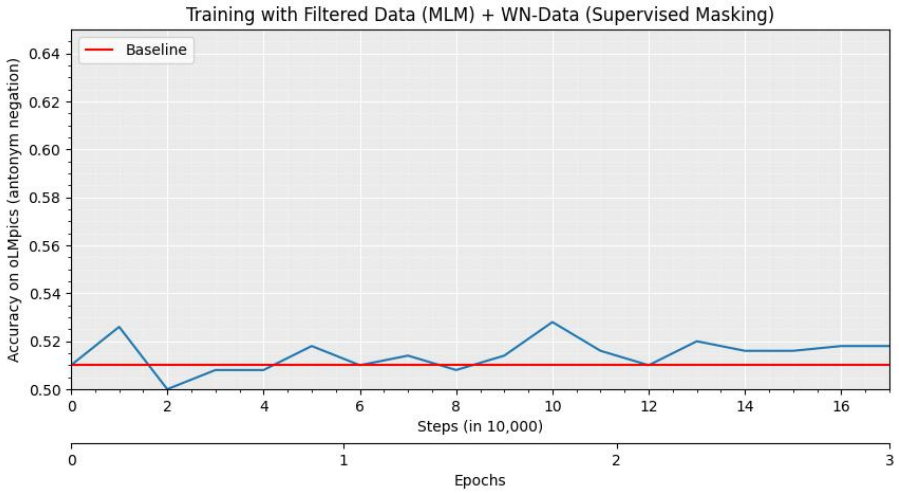

Figure 3: Experiment 2: Training BERT-Small on the filtered dataset using unsupervised MLM and the WordNet adversarial dataset using supervised masking.

A.6 Hyperparameters - Experiment 3

Parameter	Value
Model Identifier (transformers library)	`prajjwal1/bert-small`
Optimizer	AdamW
Learning Rate	5e-5
Epochs	3
Batch Size	24
Blocksize	128
Data Collator for Training	`DataCollatorForLanguageModeling` (transformer library)
Test Ds. Proportion	5
Steps (Evaluation)	10000
seed	23, 42, 7
Layers Frozen	0, 1, 2
Frozen Layers	bert.encoder.layer.{0,1,2,}.attention.self.query.weight bert.encoder.layer.{0,1,2,}.attention.self.query.bias bert.encoder.layer.{0,1,2,}.attention.self.key.weight bert.encoder.layer.{0,1,2,}.attention.self.key.bias bert.encoder.layer.{0,1,2,}.attention.self.value.weight bert.encoder.layer.{0,1,2,}.attention.self.value.bias bert.encoder.layer.{0,1,2,}.attention.output.dense.weight bert.encoder.layer.{0,1,2,}.attention.output.dense.bias bert.encoder.layer.{0,1,2,}.attention.output.LayerNorm.weight bert.encoder.layer.{0,1,2,}.attention.output.LayerNorm.bias bert.encoder.layer.{0,1,2,}.intermediate.dense.weight bert.encoder.layer.{0,1,2,}.intermediate.dense.bias bert.encoder.layer.{0,1,2,}.output.dense.weight bert.encoder.layer.{0,1,2,}.output.dense.bias bert.encoder.layer.{0,1,2,}.output.LayerNorm.weight bert.encoder.layer.{0,1,2,}.output.LayerNorm.bias

Table 8: Hyperparameters used in Experiment 3. All values that are not reported, are default values from the transformer library.

A.7 Evaluation Graph - Experiment 3

Figure 4: Experiment 3: Training BERT-Small on the filtered dataset and the WordNet adversarial dataset using unsupervised MLM. The yellow, green, and blue lines represent the different training runs using different seeds.

A.8 Hyperparameters - Experiment 3+

Parameter	Value
Model Identifier (transformers library)	`prajjwal1/bert-small`
Optimizer	AdamW
Learning Rate	5e-5
Epochs	3
Batch Size	24
Blocksize	128
Data Collator for Training	`DataCollatorForLanguageModeling` (transformer library)
Test Ds. Proportion	5
Steps (Evaluation)	10000
seed	42
Layers Frozen	0, 1, 2
Frozen Layers	bert.encoder.layer.{0,1,2,}.attention.self.query.weight bert.encoder.layer.{0,1,2,}.attention.self.query.bias bert.encoder.layer.{0,1,2,}.attention.self.key.weight bert.encoder.layer.{0,1,2,}.attention.self.key.bias bert.encoder.layer.{0,1,2,}.attention.self.value.weight bert.encoder.layer.{0,1,2,}.attention.self.value.bias bert.encoder.layer.{0,1,2,}.attention.output.dense.weight bert.encoder.layer.{0,1,2,}.attention.output.dense.bias bert.encoder.layer.{0,1,2,}.attention.output.LayerNorm.weight bert.encoder.layer.{0,1,2,}.attention.output.LayerNorm.bias bert.encoder.layer.{0,1,2,}.intermediate.dense.weight bert.encoder.layer.{0,1,2,}.intermediate.dense.bias bert.encoder.layer.{0,1,2,}.output.dense.weight bert.encoder.layer.{0,1,2,}.output.dense.bias bert.encoder.layer.{0,1,2,}.output.LayerNorm.weight bert.encoder.layer.{0,1,2,}.output.LayerNorm.bias
Amount (For WN Data Generation)	10

Table 9: Hyperparameters used in Experiment 3. All values that are not reported, are default values from the transformer library.

A.9 Hyperparameters - GLUE-Fine-Tuning

Parameter	Value
Learning Rate	2e-5
Batch Size	16
Weight Decay	0.1
Epochs	10
Warmup Ratio	0.06

Table 10: Hyperparameters used in fine-tuning selected GLUE tasks. All values that are not reported are default values from the transformer library.