Negation Understanding in Transformer Models

Candidate Number:1635483

Word Count: 6110

Essay submitted in partial fulfillment of the requirements for the Degree of

Master of Studies in Linguistics, Philology and Phonetics

University of Oxford

Trinity Term 2024

Attention is *Not* All You Need: Negation Understanding in Transformer Models

1635483

Abstract

State-of-the-art neural language models capture contextual word relationships but struggle with the sentence-level impact of negation. This paper investigates language models' understanding of negation in recognizing textual entailment (RTE). Rule-based data augmentation can make classifiers "negation-aware", improving performance; however, the complexity of natural language entailments raises questions about models' true understanding of negation. Inspired by Hosseini et al. (2021) and Helwe et al. (2022), this study explores the effectiveness of fine-tuning DistilBERT models on an RTE dataset with and without negation augmentation, analyzing specific examples to assess the impact of fine-tuning on understanding negation in natural language inference. The results suggest that "negation-aware" models attend more heavily to negation but fail to improve linguistic negation comprehension in any consistent or repeatable way.

1 Introduction

Small changes in sentence-level information can drastically alter the meaning of a sentence or even invert its truth value. Accurately recognizing and interpreting the connection between sentences requires an understanding of these changes and their contributions to underlying meaning. Capturing the impact of linguistic phenomena like negation on other elements within a sentence is key to various natural language processing (NLP) and natural language inference (NLI) tasks, such as sentiment analysis, information retrieval, and entailment classification. For example, it is trivial to native speakers of English that (1a) and (1b) have opposite meanings. From (1a), it would follow that (1c) is true, but (1b) does not make (1c) explicitly true or false, as John could abstain from meat without being a vegetarian.

(1) a. John is a vegetarian.
b. John is not a vegetarian.
c. John does not eat red meat.

State-of-the-art neural language models rely on a framework of distributional semantics, which assumes that words appearing in similar contexts have related meanings (Harris, 1954). While language models like BERT (Devlin et al., 2019) can capture contextual relationships between words using dense vector representations of meaning, they struggle to properly characterize and evaluate the sentence-level impact of negation (Ettinger, 2020; Hosseini et al., 2021). A distributional semantics fails to effectively capture the relative importance of negation to a sentence's meaning, as negated phrases often appear in similar contexts to their non-negated counterparts, differing only by the presence of the negation itself. Through rule-based data augmentation, one can improve the performance of a classifier on textual entailment and text generation by making them "negation-aware" (Anschütz et al., 2023; Helwe et al., 2022; Hosseini et al., 2021; Khandelwal & Sawant, 2020; Truong et al., 2022), but do the language models actually understand the negation as we do, or are they just arbitrarily matching patterns seen during training? Prior work demonstrating the failures of pre-trained language models (LMs) on negation has led to a variety of hypotheses, from confusion about factuality (Kassner & Schütze, 2020; Singh et al., 2023) to blatantly ignoring negation altogether (Jang et al., 2023). In this paper, I aim to investigate whether anything can be said about a model's *understanding* of negation in inference tasks after fine-tuning. I focus on recognizing textual entailment (RTE), an NLI task, often linked to natural language "reasoning", which involves determining whether a given premise sentence entails, contradicts, or is neutral to a hypothesis sentence (Dagan et al., 2005). State-of-the-art pre-trained language models can achieve up to 95% accuracy on RTE datasets, though many datasets contain homogeneous examples which lack negation (Helwe et al., 2022; Hossain et al., 2020), or are riddled with noise from human label variation (Plank, 2022). Some models have seen success with different optimization objectives and data augmentations, though it is unclear exactly how those contribute to the model's inference ability. Given the resources available, I intend to simulate the methods of Hosseini et al. (2021) and Helwe et al. (2022) and explore specific examples where the models continue to perform poorly. In the sections below, I explain why natural language negation is particularly complex, and go into detail about related work, my methods of fine-tuning DistilBERT (Sanh et al., 2019) models on a dataset with and without negation augmentation

and the various techniques I employ to analyze their outputs.

2 Negation

Negation is the expression of semantic opposition, denial, or falsity. Negation has been analyzed from many different angles, with various proof systems, semantic frameworks, and logical vocabularies. Similar to truth-conditional semantics, negation is often defined in terms of its impact on truth: the negation of a proposition is true if and only if the original proposition is false. Early analytic philosophers agreed that this definition of negation is particularly useful as a tool for reasoning and argumentation (Frege, 1919; Whitehead & Russell, 1997). Negation as a unary operator (¬) allows propositional and modal logic to express contradiction, as well as the impossibility or non-necessity of something. This logical notion of negation is often understood as a sentential operator, where "¬P" is interpreted as "it is not the case that P". However, natural language negation does not typically manifest as a clause-peripheral morpheme, except in rare cases (Horn, 1989). While logical uses of negation correspond to formal representations of semantic meaning, they may lack an explicit grounding to distinguish between a concept and its referents. In truth-conditional semantics, the meaning of a sentence is determined by the conditions under which it would be true. Therefore, a proper distinction between the specific concept and referent or set of referents is necessary (Frege, 1892). Logical negation is relevant to understanding truth-functional negation, but it leaves much to be desired when it comes to natural language. In natural language, negation is a semantic operator expressed syntactically and pragmatically in various ways (Horn, 1989). Negation interacts with quantifiers, modals, and other extra-linguistic elements of an utterance to enable a wider range of interpretations than explicitly truth-conditional ones. Interpreting natural language negation properly requires an understanding of these factors and their impact on one another. For example, negating a sentence like (2) can trigger two possible interpretations of an utterance, depending on negation scope and interactions with lexical meaning:

(2) Some cats are black.

   a. No cats are black.
   b. Most cats are not black.

In sentence (a), the existential quantifier "some" asserts that there exist cats that are black. The negation of this sentence, (a), results in a universally quantified

statement asserting that there are no cats that are black (there do not exist some black cats, or any for that matter). The negation in sentence (b) scopes below the quantifier and interacts with the predicate "are black," resulting in a statement that asserts that the majority of cats are not black, but rather some other color. The LFs in figure 1 show the difference between these two interpretations–the culprit, neg-raising, is a covert movement in which the interpretation of the negation moves above (or stays below) a quantifier to generate different meanings (Collins & Postal, 2018; Horn, 2020). Adjectival and predicate negation in (2b) demonstrate that negation can furthermore introduce ambiguity when a word does not necessarily have an opposite.

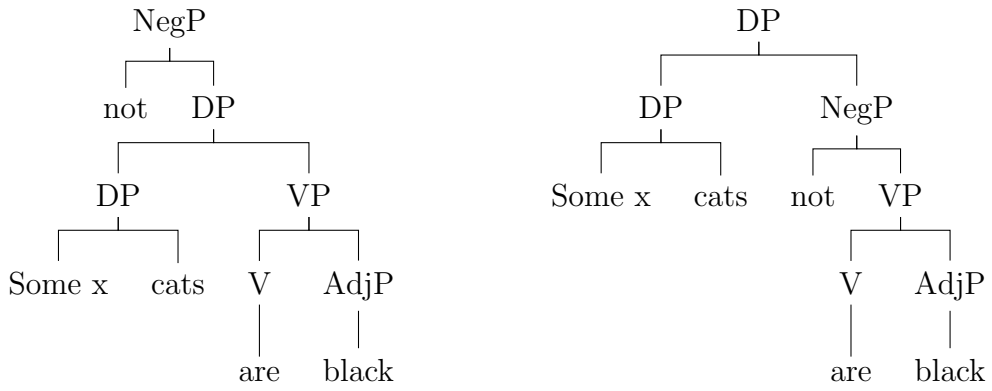

Figure 1: LFs of negated (2)

Negative statements are more morphosyntactically marked, yet less informative than their affirmative counterparts. When a proposition is negated, the existence of a more informative alternative is implicated, leading the interpreter to infer additional meaning based on context and knowledge of the world. The presence of alternatives, like any non-black color of cat from example (2), requires some connection to extra-linguistic information, as we cannot deduce the color of the cat strictly from the semantics.

Given the wide range of different interpretations of negation in natural language, it remains an open challenge in NLP to effectively capture the contribution of negation. If logical, semantic, and pragmatic features of negation are latent in language data, what exactly is learned from positional encodings and contextual embeddings? Will providing additional data with examples of negation in various contexts help a model understand when to use it? I explore these and similar questions in my experiment.

3 Related Work

3.1 Distributional Semantics

Distributional semantics has proven to be a powerful framework for representing the meaning of words based on their occurrence patterns in large text corpora. By leveraging the distributional hypothesis, that words appearing in similar contexts tend to have related meanings (Harris, 1954), neural language models learn dense vector representations of words that capture rich semantic relationships between them, called embeddings. Dense vector representations can capture semantic and syntactic relationships between words. Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and FastText (Bojanowski et al., 2017) are based on Skip-Gram or Continuous Bag of Words architectures, efficient in representing word similarity by using cosine distance between their vectors. More recent approaches like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) use large libraries of data to generate contextual embeddings, capturing additional rich semantic information from positional encodings over a given context window. A "semantic" vector space can thus be derived from contexts of use, given a large enough sample size, where representations are individual vectors encoding sentential or word meaning. Classic examples demonstrate this by showing that one can retrieve the vector representation for "queen" by taking the vector for "king", subtracting the vector for "man" and adding the one for "woman". However, representing words as single points in a high-dimensional space doesn't fully capture the subtleties and flexibility of meaning across sentences. For example, a sentence and its minimal pair negation can often appear in the same context:

(3) I went to the chemist to grab a few things...

a. They had Paracetamol!

b. They had no Paracetamol!

A simple distributional model of sentence meaning might assume that both (3a) and (3b)'s embeddings can be used in the same contexts, despite their contradictory meanings. Various papers have proposed more expressive representations of meaning through compositional tensors (Baroni et al., 2014; Coecke et al., 2010), static gaussian (Vilnis & McCallum, 2014) and posterior distributions (Bražinskas et al., 2018), set-theoretic regions (Dasgupta et al., 2022) and probabilistic predications over entity representations (Emerson, 2018; Emerson & Copestake, 2016) to enrich distributional models. While these approaches offer a

more linguistically principled way to model word meaning compared to simple vector-space representations, capturing sentence-level meaning remains challenging. In this work, I aim to explore the limits of distributional semantics in interpreting sentences with negation, even when training sets are enriched with out-of-vocabulary synthetic data. Perhaps future research could explore another model for representing meaning using one of the aforementioned techniques.

3.2 Recognizing Textual Entailment

Recognizing Textual Entailment is an NLP task developed to assess a model's ability to judge the interpretation of a sentence, given another one (Dagan et al., 2005). The first sentence, the premise, is treated as a source of factual information from which to make a judgment about the second sentence, the hypothesis. RTE is an inference task often linked to reasoning because properly judging the relationship of one sentence to another requires an understanding of the semantic content and the logical implications of both sentences. According to Montague (1970), sentence A entails sentence B if in all models in which the interpretation of A is true, the interpretation of B is also true. In other words, the truth of one sentence or proposition necessarily implies the truth of another. RTE aims to evaluate the model's ability to properly characterize this connection, though it critically does not consider the existence of alternatives, or implicatures, which can be proven false given the proper context (Grice, 1975). For further discussion refer to section 4.1.

Earlier symbolic and statistical approaches to RTE have been mostly supplanted by deep learning methods using datasets like SNLI, MultiNLI, XNLI, and SciTail (Bowman et al., 2015; Conneau et al., 2018; Khot et al., 2018; Williams et al., 2018). These datasets provide many hundreds of thousands of naturalistic human-annotated sentence pairs labeled as either entailment, contradiction, or neutral. Language models fine-tuned on such NLI datasets perform well on textual entailment tasks, ideally developing comprehensive, domain-general embeddings in the process. However, McCoy et al. (2019) found that state-of-the-art models trained on these datasets often rely on fallible syntactic heuristics, and consistently fail on examples targeting these heuristics. Hossain et al. (2020) demonstrate that negation is largely underrepresented in such datasets, and models perform poorly when new examples with negation are introduced. Hosseini et al. (2021) and Helwe et al. (2022), with this idea in mind, use principled systems to create new data from SNLI, MNLI and RTE datasets and train models based on unlikehood loss for optimization, showing some improvement. Anschütz et al. (2023) found that

prior work to advance the negation-awareness of large language models focuses mostly on model optimization and minimal work has been published on benchmarks and evaluation.

3.3 "Understanding" negation in LMs

Previous work has shown that pre-trained language models struggle with negation, leading researchers to propose various hypotheses. Some suggest that LMs perform shallow, inconsistent pattern matching in the presence of "not" (Kassner & Schütze, 2020), while others argue that LMs simply ignore negation altogether at inference (Jang et al., 2023; Singh et al., 2023). Singh et al. (2023) probe negation via masked language modeling, and report that pre-trained LMs can detect the presence of "not" up to the final layer of the network, but explicit fine-tuning on the changed outputs of negated factual propositions as done in (Hosseini et al., 2021) does not help incorporate negation into factuality judgements. They indicate that language models need more stimuli to understand negation than triggers like "not", and resolving negation is most difficult in the presence of multiple possible referents. Understanding negation may become easier with a stronger connection to real-world knowledge, though models must have an implicit mechanism for disambiguation. Somewhat related, A. Liu et al. (2023) find that ambiguity-sensitivity is severely lacking, with GPT-4 only able to disambiguate 32% of the time when compared to human evaluators, and processing underspecification remains an open challenge (Wildenburg et al., 2024; Yang et al., 2023). The literature seems to point to the fact that fine-tuning a classifier on a task does not change how an underlying pre-trained LM understands the meaning of an encoded sentence; rather, the classifier may rely on surface-level pattern matching at inference time. Fine-tuning on negation-rich data may help a model recognize the importance of negation in the context of the task, and improve its ability to interpret the relationship between sentences. My experiment aims to take into account prior work and locate examples where a classifier fine-tuned on negation succeeds and fails at classifying entailment and explore possible ways synthetic data augmentation helps at inference.

4 Experiment

Following previous work, I implement transformer-based models on a textual entailment dataset that has been augmented with negated instances. As previous papers were limited to strictly grammatical sentences, I compare performance of

the models on data deemed 80% or more grammatical by a DistilBERT (Sanh et al., 2019) classifier fine-tuned on the CoLA standards for grammaticality (Warstadt et al., 2019) to the full dataset, as well as their augmented datasets.

4.1 Data Augmention

Data augmentation is commonly used in deep learning to generate synthetic data for improved model performance. Symbolic data augmentation techniques, like feature-space augmentation (DeVries & Taylor, 2017), use rules or discrete structures to generate synthetic examples by transforming data features. This approach helps create diverse training data, reduces overfitting, and improves model generalization, especially when annotated linguistic datasets are scarce. Unlike a Montagouvian definition of entailment, NLP literature has more loosely defined entailment as a means of showing that a given text fragment is *likely* to follow from another (Dagan et al., 2005; Glickman & Dagan, 2005). This definition addresses the gradedness in a sentence's relationship to its context, though it fails to specify a clear distinction between entailments versus non-entailments if the sentences are related. A possible explanation is that models have unsatisfying adjective degree estimations, even after fine-tuning on the MNLI dataset (W. Liu et al., 2023). When sentences are distributionally related to one another, their probability of co-occurrence within a given context is high. Theoretically a model would then predict that two highly similar sentences should be "mutually entailing", though this isn't always the case. Helwe et al. (2022) amend this definition, adding that probabilistic entailments must have a significant threshold of acceptability or directional similarity so that an entailment is beyond any reasonable doubt, rather than just related. Consider the two similar sentences below:

(4) a. I won the lottery. (Helwe et al., 2022)
b. I played the lottery.

In all possible worlds where (4a) is true, (4b) is guaranteed to be true as well, suggesting that (4a) entails (4b). However, world knowledge tells us that the reverse is not necessarily true, even if the sentences are related; playing the lottery does not guarantee winning it. Thus, (4b) does not entail (4a), despite the increased probability of their co-occurrence. Unfortunately, there is no consistent threshold of acceptability for entailment versus non-entailment. Following (Anschütz et al., 2023; Helwe et al., 2022; Hosseini et al., 2021), I employ a rule-based data augmentation scheme and optimization objective function which

takes these asymmetries into account.

4.1.1 Dataset

For this experiment, I use the Multi-Genre Natural Language Inference dataset (Williams et al., 2018), a human-annotated corpus of around 400k premise-hypothesis pairs, labeled as either entailment, contradiction, or neutral. I chose MNLI for two primary reasons: it covers text of different genres, giving the model access to a range of examples, and it has longer instances, the longest input "sentence" being 384 words. Unlike other RTE datasets, MNLI contains ungrammatical examples, many of which would be trivial for a human to understand. For example:

(5) **Premise:** uh somewhat they're not my favorite team i am uh somewhat familiar with them
Hypothesis: They are the best team in the league, by they are not my favorite.
Label: neutral

Previous work only deals with strictly grammatical, single sentences under a certain length. Merrill (2022) shows how restricting sentences to purely grammatical ones loses out on common natural language redundancies or elaborations, missing potential analysis of how humans actually use language. For this reason, I've gone ahead and trained models on both the full dataset, as well as a grammatical-only dataset for comparison. The "grammatical" dataset contains only sentence pairs deemed at least 80% grammatical by a DistilBERT (Sanh et al., 2019) classifier fine-tuned on the CoLA standards for grammaticality (Warstadt et al., 2019).

4.1.2 Creating New Examples

Out of 392,702 pairs of sentences in the original training set, 97,247 (24.76%) contain a negation. Out of 237,576 grammatical pairs of sentences in the "grammatical" training set, 55,087 (23.19%) contain a negation. Using the open-source python package negate (Anschütz et al., 2023), I create balanced dependency-based verbal negation data to augment the original datasets with provable logical perturbations of each pair of input sentences (for proofs, see Helwe et al., 2022). I exclude negative examples, where the logical negation of input sentences yields a non-entailment or non-contradiction. By using only positive logical perturbations, I directly compare the performance of the augmented model

with the base model. In theory, this isolates the impact of adding negation-rich positive examples to the training data, without confounding factors like negative examples that may introduce noise. The negation step adds around 250k examples and 150k examples to the full and grammatical datasets, respectively.

4.2 Training and Testing

All models are trained on an NVIDIA A100 GPU, whose compute was split between 10-15 students at any given time. Given computational constraints, I opt for DistilBERT (Sanh et al., 2019), which is 60% smaller and more lightweight than BERT. With 6 layers and 12 attention heads, DistilBERT has 6 fewer layers than BERT while maintaining 95% performance compared to BERT on GLUE benchmarks. The models are tested on the 10k "matched" test set from MNLI, and evaluated via macro F1 score. To specifically isolate the model performance on negative samples, I compare the F1 scores of models on the whole test set (10k samples), to only those containing negation (2371 samples).

$$\text{Macro F1} = \frac{1}{N}\sum_{i=1}^{N}\frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{1}$$

where N = number of samples, and:

$$\text{Precision}_i = \frac{\text{True Positives}_i}{\text{True Positives}_i + \text{False Positives}_i} \tag{2}$$

$$\text{Recall}_i = \frac{\text{True Positives}_i}{\text{True Positives}_i + \text{False Negatives}_i} \tag{3}$$

4.2.1 Architecture

I use a transformer architecture (Vaswani et al., 2017) because they are regarded as state-of-the-art for natural language understanding tasks like RTE, their self-attention mechanism is made to handle long sequences to capture contextual information, and their embeddings in some ways already encode negation's impact on surrounding words. Transformers are widely cited, used, and even praised for their intelligence, though their capacity for understanding language seems far more limited. I selectively freeze the first five layers of DistilBERT and fine-tune the last layer for RTE to use (and prevent the model from forgetting) learned information

in pre-trained embeddings, and make training more computationally efficient, while adapting the last layer to the specific requirements of RTE. From there, the fine-tuned model can be tested on its ability to interpret negation on complex, real-world language data outside the scope of its initial pre-training.

4.2.2 Hyperparameters

I run a standard random-search hyperparameter tuning on the number of epochs, batch size and learning rate. After 20 runs, I found that performance improvements over Hossain et al. (2020)'s hyperparameters were marginal at best, and stuck with the hyperparameters given in Appendix B.

4.2.3 Loss

I employ cross-entropy loss as the objective function for training, where:

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \tag{4}$$

- N is the number of samples in the dataset, C is the number of classes.
- y_{ij} is the true label of the i-th sample for the j-th class, encoded as a one-hot vector.
- $\hat{y}_{ij}$ is the predicted probability of the i-th sample belonging to the j-th class.

The cross-entropy loss measures the dissimilarity between the predicted probability distribution $\hat{y}$ and the true probability distribution y (Nadaraya, 1964). There is a penalty for assigning high probability to incorrect classifications and a reward for assigning the correct class a high probability. While Helwe et al. (2022) utilized unlikelihood loss to additionally penalize a model for predicting incorrect labels on negative input, I opt for cross-entropy loss in an attempt to isolate only the comparable performance of augmented model and the base model. I use only provable logical examples as a small perturbation to avoid over-indexing on augmentation and inadvertently increasing noise.

Models	**Random Baseline**	**DistilBERT-base**		**DistilBERT-aug**	
		Full	Neg	Full	Neg
Fine-tune					
Full (392.7k+250k)	0.333	0.613	.610	.647	.633
Gram (237.5k+150k)	0.333	0.603	.590	.643	.636

Table 1: Macro F1 scores for different models

5 Results

5.1 Impact of Data Augmentation

Table 1 shows the macro F1 scores of each model (base and augmented) on the full and negation-only test sets when trained on the different datasets, full (Full_{base}, Full_{aug}) and grammatical (Gram_{base}, Gram_{aug}). Overall, the Full_{aug} model outperformed the Full_{base} model by 50 total negation predictions and 224 total affirmative predictions, though the set of correct predictions by the base model is not a proper subset of the augmented model's correct predictions. Likewise, Gram_{aug} outperformed the Gram_{base} by 65 negation predictions and 215 affirmative predictions. I compute a McNemar test (McNemar, 1947) on each set of model predictions to show that there is a statistically significant improvement from the data augmentation on model performance. The comparative performance of Full_{base} vs. Full_{aug} has a p-value of 4.28e-8, and Gram_{base} vs. Gram_{aug} has a p-value of 1.23e-12, both much smaller than a confidence level of 0.05, allowing us to say there's a statistically significant improvement from the augmentation. Still, there are many instances where the base models beat the augmented models, which can deceive a shallow test statistic only testing overall results.

The confusion matrices for the Full_{base} and Full_{aug} models in figure 2 show that the Full_{base} has more true positives for entailment compared to Full_{aug}, and fewer true contradictions and neutrals. The base model predicts many more false positives for entailment and false negatives for neutral and contradiction than its augmented counterpart. This means the base model has a stronger tendency to classify two sentences as entailments, while the augmented model has a more balanced distribution across true labels. Both models often confuse entailments with neutral labels, and improvements from data augmentation suggest that exposure to more negation examples during training helps the model better distinguish between entailment, contradiction, and neutral at inference. The grammatical models have a similar distribution–for those matrices, see appendix C.

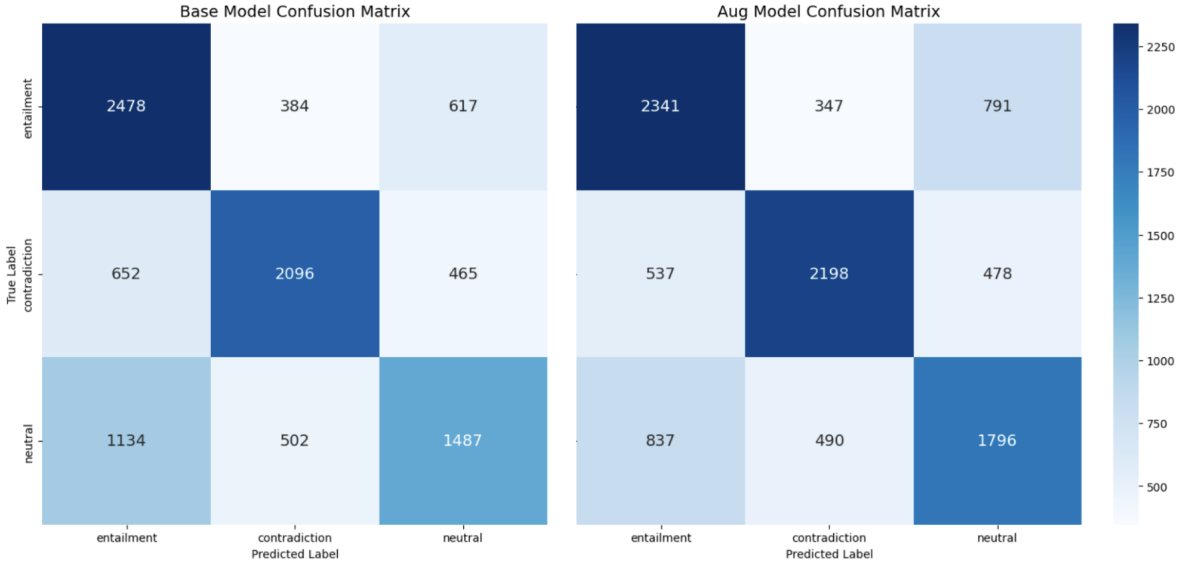

Figure 2: Full_{base} and Full_{aug} Confusion Matrices

5.2 Comparing the base models

The models trained on the full dataset outperform the grammatical-only models overall, but the improvement is negligible at 18 examples (0.18% of the test set). Grammaticality alone may not be sufficient for improving performance, or perhaps the model is not learning from grammatical sequences at all, but rather from the presence of more diverse data. The two graphs in Figure 3 compare the performance of Gram_{base} and Full_{base} on input sentences of varying lengths to see where the models differ.

Interestingly, the grammatical model is slightly better at interpreting longer sequences (>30 words) in the test set. In the graph on the right, labeled "Base Model wrong, GramBase Model right", we see that for Sentence 1 (represented by the blue bars), the number of cases where Full_{base} is incorrect but Gram_{base} is correct tends to increase slightly as the premise sentence length grows. This suggests that as the premise sentence gets longer, Full_{base} is more prone to errors whereas Gram_{base} makes an otherwise correct prediction. Perhaps as the data scales, Gram_{base} would be better equipped to utilize the additional context provided by longer input sentences, though the results from this experiment are not significant enough to show a major difference. A quick audit of the sentence length distribution of the training sets indicates that the grammatical model sees a higher relative representation of longer sequences in training, so it is more likely that the model matches the distribution of its training data. It could also be that an 80% threshold for grammaticality is not high enough to meaningfully change

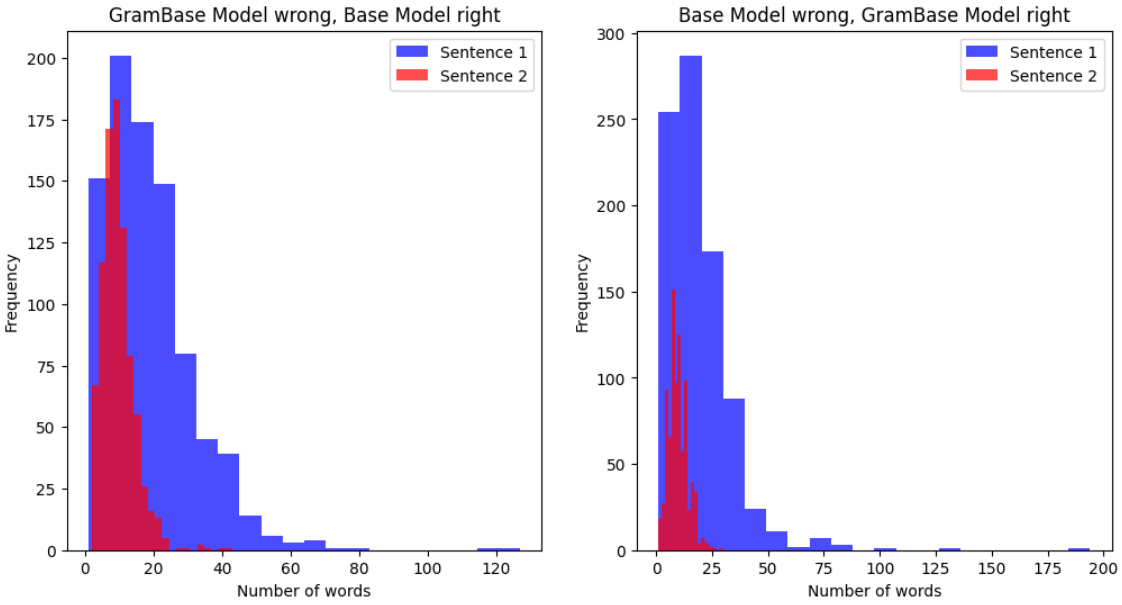

Figure 3: Gram_{base} vs Full_{base} word lengths

the training set. The main takeaway here is that more diverse data equals better coverage. As the confusion matrices for both the gram and full models share a similar distribution, I will draw from the Full_{base} and Full_{aug} models for the negation-specific test data and qualitative analysis.

5.3 Negation-only test

Do models also mirror their training distribution for classifying sentences with negation? As the dataset has a 3:1 representation of affirmative samples to samples with negation, I separate out the 97,247 and 155,248 samples with negation from the full and augmented train sets as well as the 2,371 samples from the test set for further study. In the original train and test sets, samples with negation are heavily skewed towards contradiction, with half of the samples contradiction and half of the samples about evenly spread between entailment or neutral. The augmented training set balances this spread, increasing the overall quantity of entailment and neutral data with negation.

The results from the negation-only probe show that the models mostly treat these inputs the same as affirmative inputs. As above, Full_{aug} outperforms Full_{base} on data with negation, making 174 correct predictions where Full_{base} is wrong, though Full_{base} correctly classifies 124 instances where Full_{aug} misclassifies. It is clear that the data augmentation improves the model's ability to classify contradiction and neutral, but sacrifices performance on entailment. Similar to the affirmative data,

models struggle to distinguish between entailment and neutral labels in the presence of negation. To better understand model predictions in the presence of negation, I examine model behavior when negation appears in different parts of the input sentences.

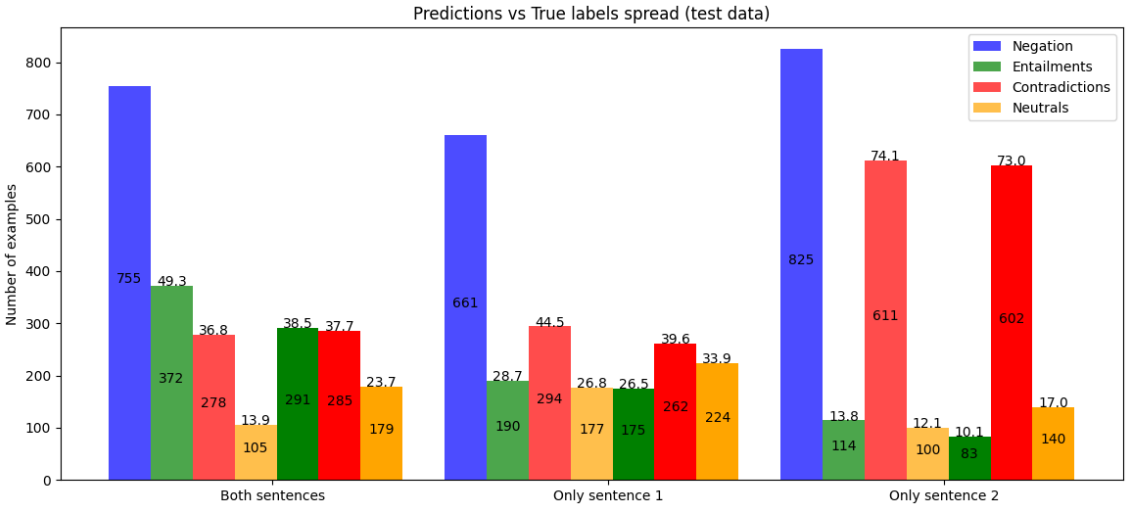

Figure 4: Full$_{base}$ predictions on Test$_{neg}$

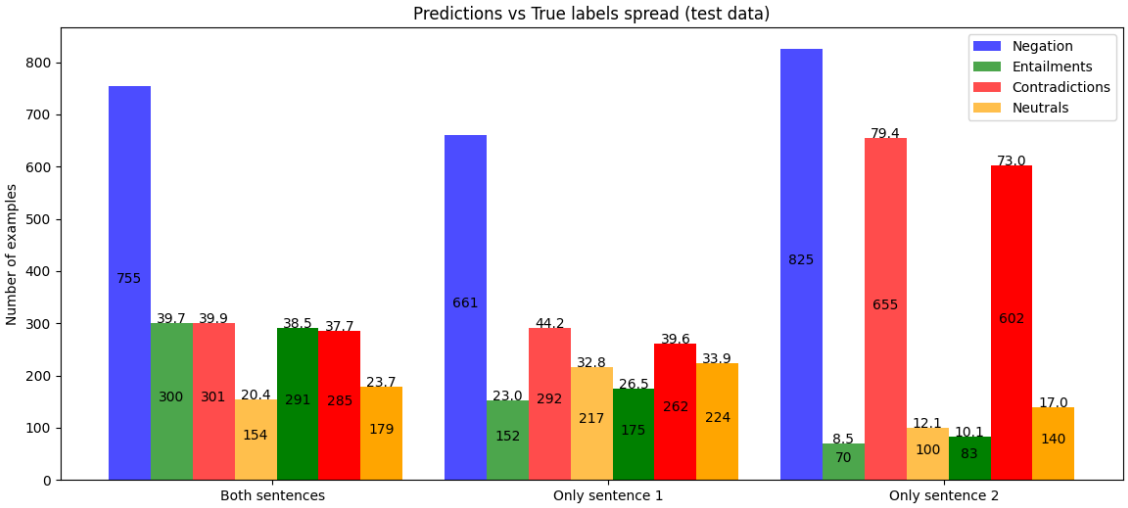

Figure 5: Full$_{aug}$ predictions on Test$_{neg}$

Figures 4 and 5 show the predicted labels of each model and the true labels of the test set when a negation appears in both the premise and hypothesis, or only one of them. The darker boxes are true labels, the lighter boxes are predictions. When Full$_{base}$ detects a negation in both input sentences, it often misclassifies the relationship as entailment. When there is a negation in either the premise or the hypothesis, the model overclassifies this as contradiction. It appears that Full$_{base}$

treats negation as a logical operator or logical cue on top of the content of the sentences, inverting its truth value. Full_{aug} has a more even spread when both premise and hypothesis contain a negation, but significantly over-indexes on contradiction when either the premise or hypothesis contain a negation. I test cosine similarity of the two input sentences and confirm that cosine similarities of all labels have significant overlap, indicating that similarity alone bears no outsized impact on a pair's true label. For these distributions, see appendix D. As the model prediction similarities do not deviate much from the true labels, the model likely makes inferences based on other features. I explore these in detail in the next section.

6 Discussion

Augmenting the dataset with additional positive samples that include negation improves the model performance by a significant margin, though it is difficult to see any major improvement on negation specifically. To better understand the models' predictions and see how negation interacts with other elements, I manually inspect examples where both models succeed and fail to accurately classify the relationship between the premise and hypothesis, as well as instances where there is a model differential, i.e., cases where one model makes a correct prediction while the other makes an incorrect prediction. Given prior research on negation understanding in LMs as well as my results, I suspect that Full_{base} treats negation as a sentence-level logical operator, unable to discriminate specific constituents to which it should apply. I hypothesize Full_{aug} recognizes that negation is more ambiguous, having more negation-rich entailment and neutral data in its training set. Using a loss-based saliency mapping technique, I can visualize the impact of a given word on the model's classification probability distribution and explain which tokens are most relevant to a model's prediction (Bastings & Filippova, 2020). The loss is directly relevant to the classification task at hand, rather than the gradients, as they would be in token or visual prediction tasks (Simonyan et al., 2013). This gives us a sense of explainability for how each token relatively impacts the model prediction. Though it can capture the relative difference of masking individual tokens, this method of saliency mapping is unable to capture all possible permutations of token masks. This means that while we can capture word-level impacts on a classification, simulating a "scope" of dependencies for each token is more difficult. When combined with attention, we can see which other elements a salient token attends to and get a better understanding of this relationship.

Classifying simple sentences is no problem for either model, often when constituents are flipped, and the verb is changed:

(6) **Premise:** The sacred is not mysterious to her.
Hypothesis: The woman is familiar with the sacred.
Both models (true label): entailment

In the underlying DistilBERT embeddings, "not mysterious" is somewhat close to "familiar"; the attention layers in both models show that they can easily discern that the familiarity in question is related to constituents the woman ("her"), the sacred and not mysterious. Taking a look at the saliency map in 6, the NP constituents have the same relative importance in both premise and hypothesis, while the verbs are most important in the classification. Ostensibly, the model weighs the decision based on the adjectival difference after seeing that the NPs are the same.

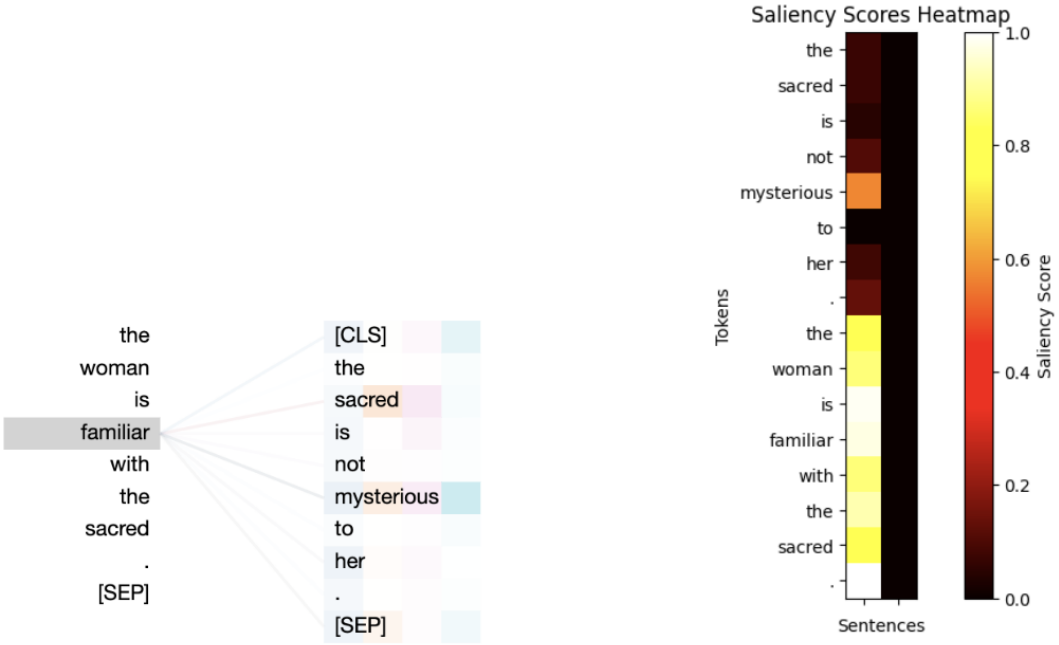

Figure 6: Attention and Saliency for (6)

Many of the cases where both models make incorrect predictions contain long premise and hypothesis pairs, where the scope and relevance of certain tokens to others are more ambiguous. This indicates that referent ambiguity and complex sentences make interpretation difficult, regardless of whether a sentence contains a negation or not. In (7), the negation of the hypothesis should only be relevant to the first conjunct in the premise, though the attention weights show that "not"

attends almost equally to both conjuncts in the premise, as does "denies" during self-attention. This means that the models think that both conjuncts are equally important enough to deem the single-conjunct hypothesis neutral rather than contradiction or entailment.

(7) **Premise:** The employee was later rehired, and Bob denies the charge.
Hypothesis: The employee did not get their job back
Both models: neutral
True label: contradiction

The saliency maps for the base and augmented models are similar, showing that the models get distracted by the negation verb "denies" from the second conjunct of the premise rather than focusing on the important information about the employee's job. A BERT model pools the meaning of an input sentence in the [CLS] token, meaning that it does not create separate representations of meaning for different conjuncts–"denies" will be considered a negation verb of the whole premise.

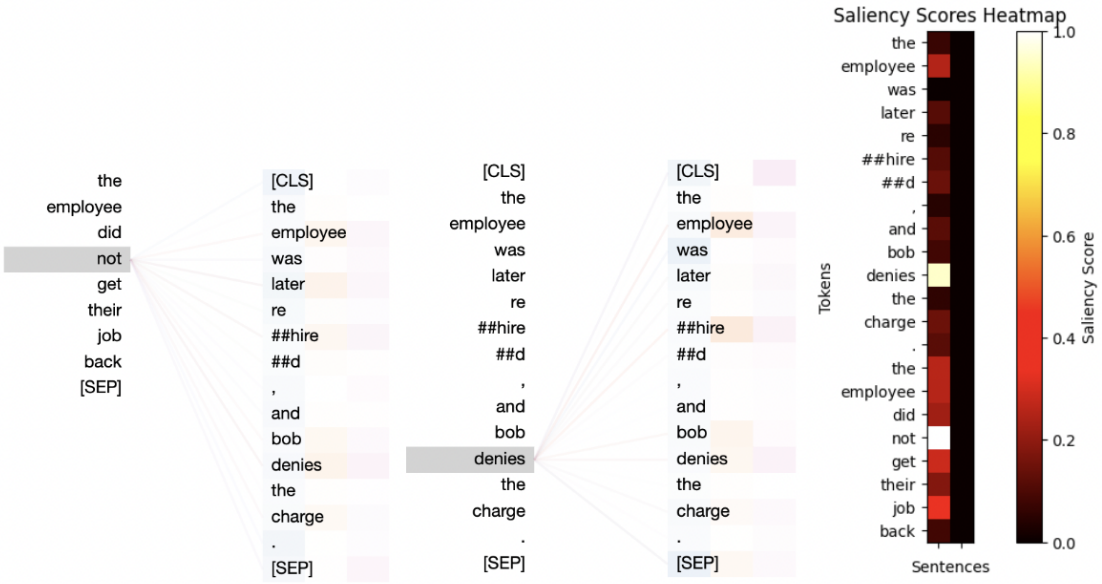

Figure 7: Attention, Self-Attention and Saliency for (7)

Data augmentation seems to help in examples where natural language negation serves a pragmatic function in the premise rather than a truth-conditional one. In (8), Full$_{base}$ still treats negation as a logical operator, finding the premise and hypothesis enough of a match to be entailment. Full$_{aug}$, on the other hand, recognizes that "no" in the first sentence is not used logically, and doesn't reverse the meaning of the predicate. The saliency map of Full$_{aug}$ considers "didn't" more

important in the final classification, while Full$_{base}$ weighs only the predicate as important. For the saliency maps, see appendix E

(8) **Premise:** no i i just painted
Hypothesis: I didn't just paint.
Full$_{base}$ (wrong): entailment
Full$_{aug}$ (true): contradiction

Though this would be a nice explanation for the impact of data augmentation, these improvements are likely coincidental. Other examples confirm that the opposite is also true; the augmented model over-indexes on contradictions in light of seeing a negation in only one of the sentences (in line with the findings of 5), even when it applies to a categorical opposite of the premise. In (9), both "all" and "none" are functionally opposite, as are "purebreds" and "mutts". A human would have no problem understanding that if all of a set of dogs are purebreds, then none of them are mutts.

(9) **Premise:** oh yeah all mine are uh purebreds so i keep them in
Hypothesis: none of mine are mutts
Full$_{aug}$ (wrong): contradiction
Full$_{base}$ (true): entailment

The attention weights show that the underlying model correctly attends to "mine", "none", "purebreds" and "mutts" to try and relate the two sentences. Saliency maps (appendix E) show the difference between the two classifiers. Full$_{aug}$ overemphasizes the importance of the negative determiner "none" without properly considering the content it negates. On the other hand, the Full*base* classifier recognizes the relevance of "mutts" but seems confused by other tokens, failing to give appropriate weight to the negative determiner. If the premise sentence were instead "all of mine are purebreds", Full$_{aug}$ still predicts a false contradiction. The selected examples demonstrate the model's understanding of lexical negation but failure to see and apply sentence-level negation scope. After seeing where the models fail, it is clear they have some idea as to how negation impacts a sentence, but it is incomplete. The marginal improvements to the performance on the original test set likely come from the addition of more data, rather than an improved understanding of linguistic negation. Especially since the data augmentation adds logical negations to the training set, it would not make sense for the model to learn pragmatic negation from only logical synthetic data. Examples (7), (8) and (9) show that referent ambiguity indeed poses a challenging

problem for the interpretation of complex sentences (Singh et al., 2023). The experiment's results demonstrate that while the data augmentation helps the model attend to negation, negation "awareness" is not negation understanding. Neural language models learn to predict patterns from their training set, with limited ability to generalize. Full_{aug}'s performance improvement on contradiction and neutral labels in the presence of negation indicates that exposure to more diverse negation examples during training can help the model better distinguish between these labels at inference time. The weaker performance on entailment labels suggests that a logical data augmentation does not teach the model the meaning of negation, just that negation is a linguistic feature that manifests in many different ways. True improvements in human-like understanding require more sophisticated and explicit representations of negation scope, focus, and interaction with other linguistic phenomena.

7 Limitations

Given computational constraints, I elected to use an efficient, rule-based negation scheme as described in Anschütz et al. (2023). Manual inspection of augmented sentences in the original dataset revealed shortcomings in the negation process consistent with Helwe et al. (2022), particularly for longer and more complex premises. It is difficult to properly negate a sentence with multiple conjuncts, or embedded clause(s). The python package used to create negated data only creates verbal negations and cannot properly handle negative polarity items (NPIs), affixal negations, negative pronouns or adverbs. It would be much easier to isolate the model's understanding of negation with a dataset that contained a spectrum of sentences and their negated pairs, but this is the best open-source option to create one. That said, it mirrors the efforts of previous papers in creating verbal negations (Helwe et al., 2022; Hosseini et al., 2021). The goal of this paper was to see exactly where and how state-of-the-art "negation-aware" transformer models learn from synthetic negation data, which was accomplished.

The models were only run on one architecture, a transformer with a specific configuration. More tests could be run with a suite of models, including more recent models as well as those incorporating methods mentioned in section 3.1, to see how different architectures interpret linguistic information. Another limitation, particularly with the MNLI dataset, is in the annotator labels. As described earlier, entailment relationships are tricky to define and possibly ambiguous in the presence of quantification or scalar adjectives. For example:

(10) **Premise:** i know people who eat tons of unhealthy food and they're just as healthy as can be
Hypothesis: Some people who eat unhealthy foods are not sick
Full$_{aug}$ (wrong): neutral
Full$_{base}$ (true): entailment

The true label in the test set is entailment, when I would argue that an example like this is neutral, because it is an implicature. Perhaps these people who eat unhealthy foods are sick–they're as healthy as can be, but not healthy by universal health standards. A solution here would be to say that sick and healthy are logical opposites, but this is not necessarily true. This is one case among many where a linguistic negation introduces many possible alternatives.

8 Conclusion

A "negation-aware" language model fine-tuned on a negation-rich dataset does not understand negation any better than the underlying model, but rather learns a more complex heuristic function in the presence of more data. That said, LMs do not understand negation in general and often make arbitrary decisions based on what they deem important (which is black-box). Models seem to use negation as a logical operator with little regard for the deeper pragmatic meaning that negation offers in natural language. While logical, semantic, and pragmatic features of negation are latent in language data, positional encodings and contextual embeddings alone are not sufficient for a model to fully understand the complexities of negation in various contexts. I concur that language models need more stimuli to understand negation. However, the foundations of word and sentence representations in large LMs prevent this without access to an infinitely large dataset. Such a dataset would need to cover all possible permutations of how negation and language are used.

References

Anschütz, M., Lozano, D. M., & Groh, G. (2023). *This is not correct! negation-aware evaluation of language generation systems.*

Baroni, M., Bernardi, R., & Zamparelli, R. (2014). Frege in space: A program for composition distributional semantics. *Linguistic Issues in Language Technology*, *9*.

Bastings, J., & Filippova, K. (2020, November). The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In A. Alishahi, Y. Belinkov, G. Chrupała, D. Hupkes, Y. Pinter, & H. Sajjad (Eds.), *Proceedings of the third blackboxnlp workshop on analyzing and interpreting neural networks for nlp* (pp. 149–155). Online: Association for Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, *5*, 135–146.

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Bražinskas, A., Havrylov, S., & Titov, I. (2018, August). Embedding words as distributions with a Bayesian skip-gram model. In E. M. Bender, L. Derczynski, & P. Isabelle (Eds.), *Proceedings of the 27th international conference on computational linguistics* (pp. 1775–1789). Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Coecke, B., Sadrzadeh, M., & Clark, S. (2010). Mathematical foundations for a compositional distributional model of meaning. *arXiv preprint arXiv:1003.4394*.

Collins, C., & Postal, P. M. (2018). Disentangling two distinct notions of neg raising. *Semantics and Pragmatics*, *11*, 5–EA.

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S. R., Schwenk, H., & Stoyanov, V. (2018). Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 conference on empirical methods in natural language processing.* Association for Computational Linguistics.

Dagan, I., Glickman, O., & Magnini, B. (2005). The pascal recognising textual entailment challenge. In *Machine learning challenges workshop* (pp. 177–190).

Dasgupta, S., Boratko, M., Mishra, S., Atmakuri, S., Patel, D., Li, X., & McCallum, A. (2022, May). Word2Box: Capturing set-theoretic semantics of words using box embeddings. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 2263–2276). Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *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)* (pp. 4171–4186). Association for Computational Linguistics.

DeVries, T., & Taylor, G. W. (2017). Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*.

Emerson, G. (2018). *Functional distributional semantics: Learning linguistically informed representations from a precisely annotated corpus* (Unpublished doctoral dissertation). University of Cambridge.

Emerson, G., & Copestake, A. (2016, August). Functional distributional semantics. In P. Blunsom et al. (Eds.), *Proceedings of the 1st workshop on representation learning for NLP* (pp. 40–52). Association for Computational Linguistics.

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.

Frege, G. (1892). On sense and reference. *Translations from the philosophical writings of Gottlob Frege*, *2*, 56–85.

Frege, G. (1919). Die verneinung. eine logische untersuchung. *Beiträge zur Philosophie des deutschen Idealismus*, *1*, 143–157. (Reprinted in translation in Peter T. Geach and Max Black (eds.), Translations from the Philosophical Writings of Gottlob Frege, 1952, 117–135.)

Glickman, O., & Dagan, I. (2005, June). A probabilistic setting and lexical coocurrence model for textual entailment. In B. Dolan & I. Dagan (Eds.), *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment* (pp. 43–48). Ann Arbor, Michigan: Association for Computational Linguistics.

Grice, H. P. (1975). Logic and conversation. In *Speech acts* (pp. 41–58). Brill.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2-3), 146–162.

Helwe, C., Coumes, S., Clavel, C., & Suchanek, F. (2022, December). TINA: Textual inference with negation augmentation. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Findings of the association for computational linguistics: Emnlp 2022* (pp. 4086–4099). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Horn, L. (1989). *A natural history of negation*. University of Chicago Press.

Horn, L. (2020). Neg-raising. In V. Déprez & M. T. Espinal (Eds.), *The oxford handbook of negation.* Oxford University Press, USA.

Hossain, M. M., Kovatchev, V., Dutta, P., Kao, T., Wei, E., & Blanco, E. (2020, November). An analysis of natural language inference benchmarks through the lens of negation. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 9106–9118). Association for Computational Linguistics.

Hosseini, A., Reddy, S., Bahdanau, D., Hjelm, R. D., Sordoni, A., & Courville, A. (2021). *Understanding by understanding not: Modeling negation in language models.* Association for Computational Linguistics.

Jang, J., Ye, S., & Seo, M. (2023). Can large language models truly understand prompts? a case study with negated prompts. In *Transfer learning for natural language processing workshop* (pp. 52–62).

Kassner, N., & Schütze, H. (2020, July). Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7811–7818). Association for Computational Linguistics.

Khandelwal, A., & Sawant, S. (2020, May). NegBERT: A transfer learning approach for negation detection and scope resolution. In N. Calzolari et al. (Eds.), *Proceedings of the twelfth language resources and evaluation conference* (pp. 5739–5748). European Language Resources Association. Retrieved from `https://aclanthology.org/2020.lrec-1.704`

Khot, T., Sabharwal, A., & Clark, P. (2018). Scitail: A textual entailment dataset from science question answering. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 32).

Liu, A., Wu, Z., Michael, J., Suhr, A., West, P., Koller, A., ... Choi, Y. (2023). *We're afraid language models aren't modeling ambiguity.*

Liu, W., Xiang, M., & Ding, N. (2023). Adjective scale probe: can language models encode formal semantics information? In *Proceedings of the aaai conference on artificial intelligence* (Vol. 37, pp. 13282–13290).

McCoy, T., Pavlick, E., & Linzen, T. (2019, July). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 3428–3448). Association for Computational Linguistics.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, *12*(2), 153–157.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, *26*.

Montague, R. (1970). Universal grammar. *1974*, 222–46.

Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability & Its Applications*, *9*(1), 141–142.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1532–1543).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018, June). Deep contextualized word representations. In M. Walker, H. Ji, & A. Stent (Eds.), (pp. 2227–2237). Association for Computational Linguistics.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Singh, S., Goel, S., Vaduguru, S., & Kumaraguru, P. (2023). *Probing negation in language models.*

Truong, T., Baldwin, T., Cohn, T., & Verspoor, K. (2022, July). Improving negation detection with negation-focused pre-training. In M. Carpuat, M.-C. de Marneffe, & I. V. Meza Ruiz (Eds.), *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 4188–4193). Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vilnis, L., & McCallum, A. (2014). Word representations via gaussian embedding. *arXiv preprint arXiv:1412.6623*.

Warstadt, A., Singh, A., & Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, *7*, 625–641.

Whitehead, A. N., & Russell, B. (1997). *Principia mathematica to* 56* (Vol. 2). Cambridge University Press.

Wildenburg, F., Hanna, M., & Pezzelle, S. (2024). Do pre-trained language models detect and understand semantic underspecification? ask the dust! *arXiv preprint arXiv:2402.12486*.

Williams, A., Nangia, N., & Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 1112–1122). Association for Computational Linguistics. Retrieved from `http://aclweb.org/anthology/N18-1101`

Yang, S., Chen, F., Yang, Y., & Zhu, Z. (2023). A study on semantic understanding of large language models from the perspective of ambiguity resolution. In *Proceedings of the 2023 international joint conference on robotics and artificial intelligence* (pp. 165–170).

A GitHub Repository

The code and resources for this paper are available in an anonymous repository at https://anonymous.4open.science/r/negation-understanding/.

B Model Training Configurations

DistilBERT	**Full**$_{base}$	**Full**$_{aug}$	**Gram**$_{base}$	**Gram**$_{aug}$
Epochs	3	3	3	3
Batch Size	32	32	32	32
Learning Rate	2e-5	2e-5	2e-5	2e-5

Table 2: Hyperparameters used in training

C Gram Confusion Matrices

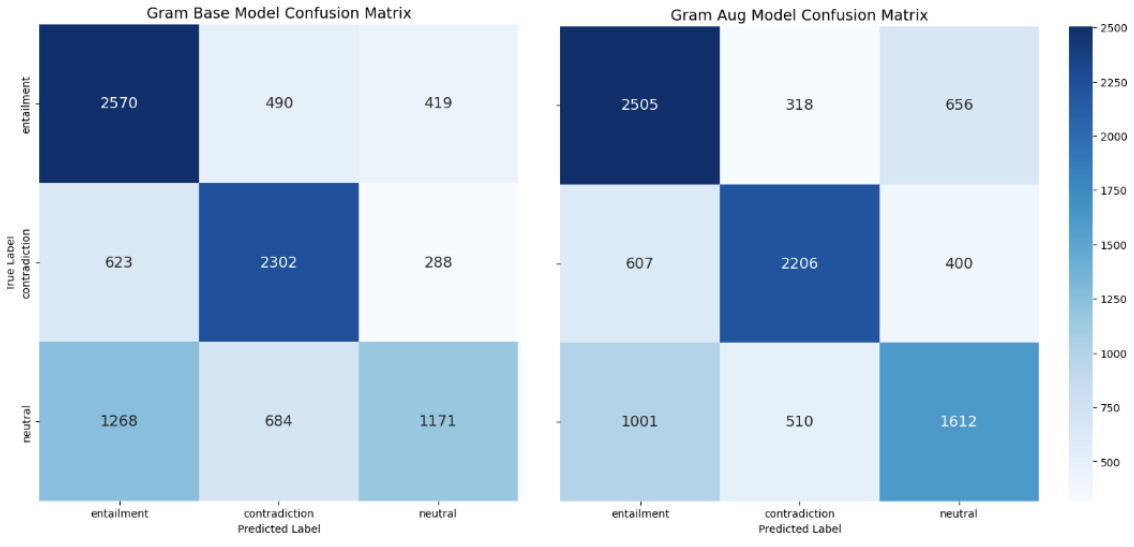

Figure 8: Gram$_{base}$ and Gram$_{aug}$ Confusion Matrices

D Input Cosine Similarities

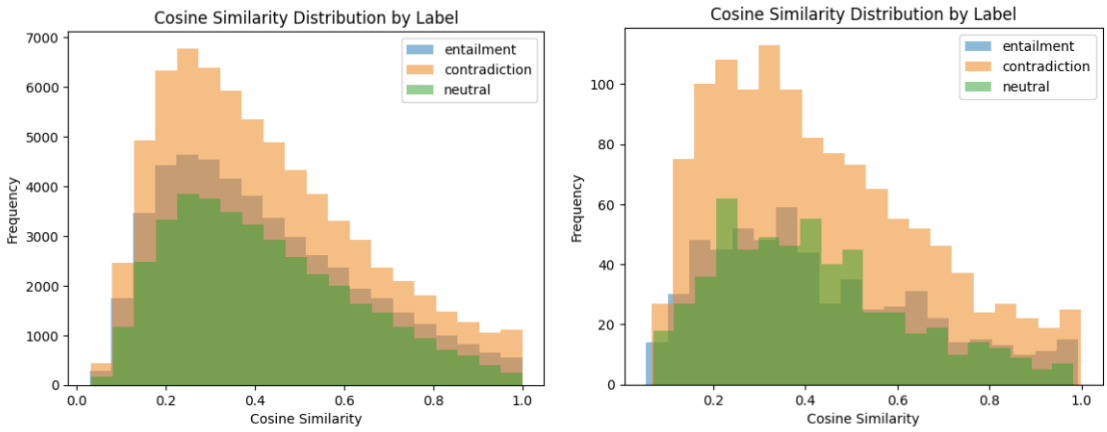

Figure 9: Augmented Dataset and Neg Test Set Cosine Similarity Distributions

E Saliency Maps

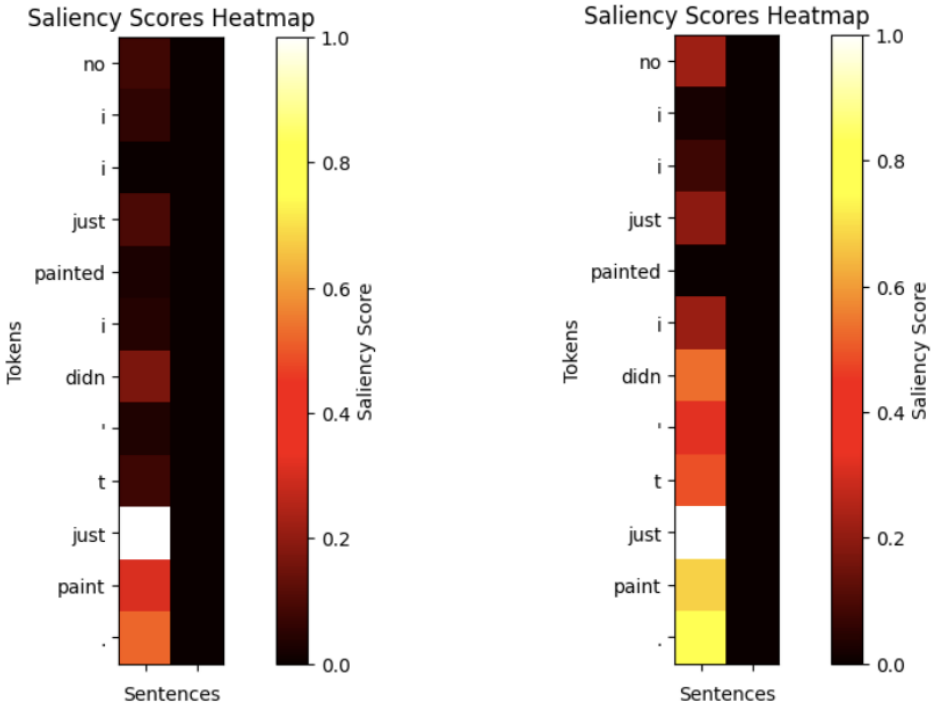

Figure 10: Full_{base} and Full_{aug} Saliency map for (8)

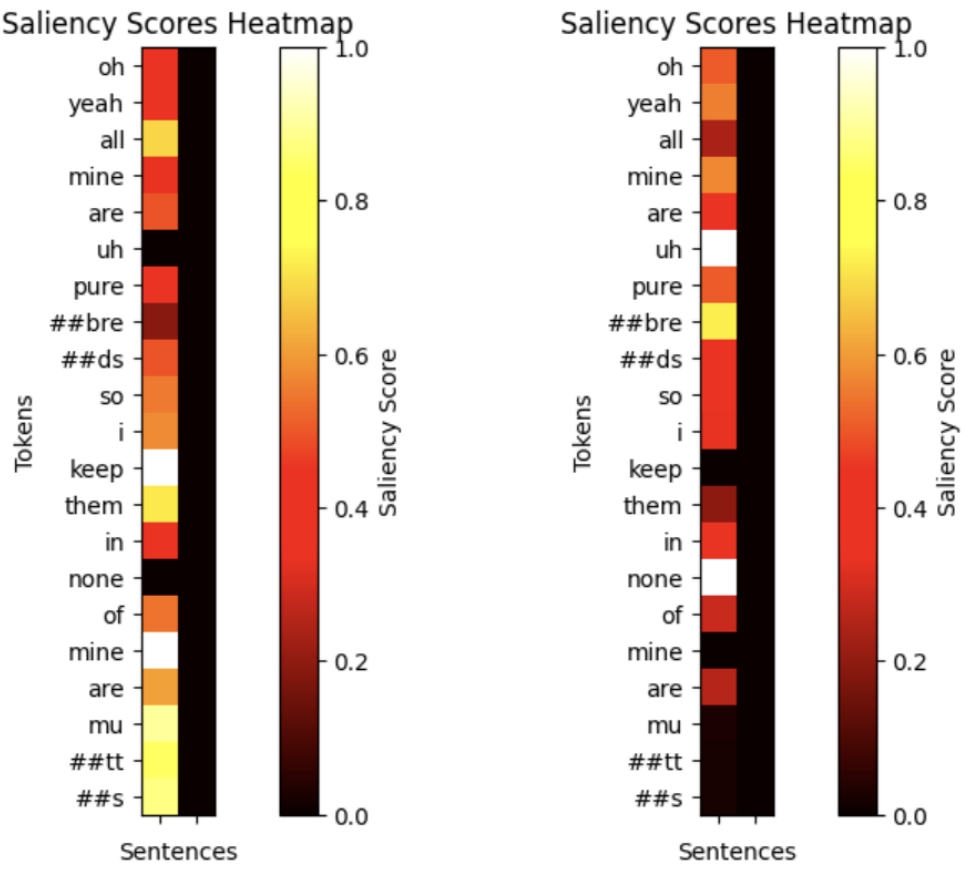

Figure 11: Full_{base} and Full_{aug} Saliency map for (9)