TINA: Textual Inference with Negation Augmentation

Chadi Helwe, Simon Coumes, Chloé Clavel, Fabian Suchanek
Télécom Paris, Institut Polytechnique de Paris, France
{chadi.helwe, simon.coumes, chloe.clavel, suchanek}@telecom-paris.fr

Abstract

Transformer-based language models achieve state-of-the-art results on several natural language processing tasks. One of these is *textual entailment*, i.e., the task of determining whether a premise logically entails a hypothesis. However, the models perform poorly on this task when the examples contain negations. In this paper, we propose a new definition of textual entailment that captures also negation. This allows us to develop TINA (Textual Inference with Negation Augmentation), a principled technique for negated data augmentation that can be combined with the unlikelihood loss function. Our experiments with different transformer-based models show that our method can significantly improve the performance of the models on textual entailment datasets with negation – without sacrificing performance on datasets without negation.

1 Introduction

Textual entailment (TE, also called Natural Language Inference) is the task of recognizing whether one natural language sentence (the *premise*) semantically entails another one (the *hypothesis*). For example, the premise "I live in Paris" entails the hypothesis "I live in France". TE is at the heart of natural language understanding, as it is closely related to question answering and natural language reasoning (Dagan et al., 2005; Poliak, 2020). Nowadays, the state of the art performance in TE is achieved by transformer-based models such as BERT (Devlin et al., 2019).

However, transformer-based models can get derailed easily by trap words or syntactic variations (see, e.g., Helwe et al. (2021) for a survey). In particular, such models have difficulties with negation in textual entailment (Hossain et al., 2020; Hosseini et al., 2021). Here is an example from Hossain et al. (2020)'s dataset:

Premise:	Green cards are **not** becoming more difficult to obtain.
Hypothesis:	Green card is now difficult to receive.
BERT Prediction:	Entailment
Label:	Not Entailment

In this paper, we provide a principled analysis of negation in textual entailment. In particular, we propose a probabilistic definition of entailment that can capture also negation. This allows us to develop TINA (Textual Inference with Negation Augmentation), an approach to automatically augment TE training datasets with negated instances. TINA uses logical deduction to generate new negated training examples from existing ones. For example, we can generate that "I don't live in France" entails "I don't live in Paris". We can then show that models finetuned on our augmented datasets are more resilient to negation, especially when combined with the unlikelihood loss. At the same time, the finetuned models perform just as well on datasets without negation. The contributions of our paper are as follows:

- a novel probabilistic definition of entailment that considers also negation;
- provably correct rules to derive new entailment relationships;
- a method to automatically augment TE datasets using these derivations;
- experiments showing that models that are finetuned on the augmented datasets are more resilient to negation in TE.

The rest of the paper is organized as follows. In Section 2, we review the related work. Section 3 describes TINA, our approach to defining textual entailment, and to making transformer-based models robust to negation in textual entailment. In Section 4, we evaluate our approach on different datasets. We conclude in Section 5, and list limita-

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4086–4099
December 7-11, 2022 ©2022 Association for Computational Linguistics

tions of our approach afterwards.

Appendix A contains the proofs of correctness, Appendix B contains the the hyperparameters used in our experiments, Appendix C shows a graphical representation of our evaluation results, and Appendix D contains a supplementary table of derivations. All data and code is available on GitHub[1].

2 Related Work

2.1 Negation in Language Models

Transformer-based models such as BERT (Devlin et al., 2019) achieve state-of-the-art results on a broad range of different NLP tasks, including machine translation, named entity recognition, and recognizing textual entailment. However, one of the pitfalls for such models is negation (Ettinger, 2020; Helwe et al., 2021; Kassner and Schütze, 2020). As shown by Kassner and Schütze (2020) and Ettinger (2020), a pretrained BERT-based model cannot differentiate between affirmative and negative statements. In addition, Niven and Kao (2019) have found that a finetuned BERT relies on simple cue words such as "not", and can thus be misled. To the best of our knowledge, the only attempt to improve the robustness of language models to negation is BERTNOT (Hosseini et al., 2021), a BERT-based model that adopts an unlikelihood objective function during training for the task of language modeling to learn to differentiate between affirmative and negative sentences.

2.2 Data Augmentation

Data augmentation is a technique to automatically create new instances in order to increase the size of a training dataset. It can mitigate problems of low-resource languages, class imbalance, and bias in datasets. Data augmentation techniques can be categorized into rule-based approaches, model-based approaches, and example interpolation (Feng et al., 2021). We are interested here in the rule-based category, which uses predefined rules to generate new instances (Hariharan and Girshick, 2017; Schwartz et al., 2018; Paschali et al., 2019; Wei and Zou, 2019; Xie et al., 2020; Şahin and Steedman, 2018; Wang et al., 2022). Our approach is inspired by the work of Wang et al. (2022), which uses logical rules for data augmentation. We go further by logically deriving new rules for data augmentation, and by combining the data augmentation with the unlikelihood loss for finetuning transformer-based models.

[1] https://github.com/ChadiHelwe/TINA

2.3 Textual Entailment

Textual Entailment is a task that was created to evaluate the "understanding capabilities" of NLP systems. The goal of this task is to determine if a hypothesis can be inferred from a premise (Dagan et al., 2005; Poliak, 2020). Different textual entailment datasets have been proposed. The most popular ones are SNLI (Stanford Natural Language Inference) (Bowman et al., 2015), MNLI (Multi-Genre Natural Language Inference) (Williams et al., 2018), and Pascal RTE (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007, 2008; Bentivogli et al., 2009).

SNLI is a large human-annotated corpus consisting of over 550K premise-hypothesis pairs that are labeled with one of the following classes: entailment, contradiction, and neutral. The premises of this dataset are image captions from Flickr30k, while its hypotheses were generated by human annotators. Here is an example from the SNLI dataset:

Premise:	A smiling costumed woman is holding an umbrella.
Hypothesis:	A happy woman in a fairy costume holds an umbrella.
Label:	Neutral

MNLI is a large dataset of around 433K instances that are labeled in the same way as SNLI. However, unlike SNLI, MNLI covers different text genres such as fiction, telephone speech, and letters, and has longer instances. It also has a large portion of less grammatical text, as in this example:

Premise:	yes now you know if if everybody like in August when everybody's on vacation or something we can dress a little more casual or
Hypothesis:	August is a black out month for vacations in the company.
Label:	Contradiction

RTE is much smaller than SNLI and MNLI, with around 5K premise-hypothesis pairs. Different from the other datasets, it has just two classes, entailment and non-entailment. Here is an instance:

Premise: Valero Energy Corp., on Monday, said it found "extensive" additional damage at its 250,000 -barrel-per-day Port Arthur refinery.
Hypothesis: Valero Energy Corp. produces 250,000 barrels per day.
Label: Entailment

The state of the art achieves an accuracy of around 92-95% on these datasets. The best models are EFL (Wang et al., 2021) for SNLI, T5-11B (Raffel et al., 2020) for MNLI , and Google's Pathways Language Model (PaLM) (Chowdhery et al., 2022) for RTE.

2.4 Negated Textual Entailment

The good performance of language models on textual entailment datasets raises the question of whether this good performance persists in the presence of negation (Hossain et al., 2020, 2022). Negation is generally underrepresented in TE datasets (Hossain et al. (2020)), with 7.16% of SNLI's sentences containing a negation, 22.63% in MNLI, and 1.19% in RTE. Therefore, Hossain et al. (2020) created new benchmarks by taking instances from SNLI, MNLI, and RTE and introducing a negation. They showed that language models perform poorly on these datasets. Hosseini et al. (2021) introduced the previously mentioned BERTNOT model to improve performance. In our work, we will show how that performance can be improved even further by using a principled way to augment the training datasets.

3 Our Approach: TINA

TINA (Textual Inference with Negation Augmentation) is our proposed approach to build a language model that is robust to negation in textual entailment tasks. Our main idea is to finetune transformer-based models on a textual entailment dataset that has been augmented with negated instances. For this purpose, let us first revisit the definition of entailment.

3.1 Defining Entailment

We say that a text fragment A *entails* a text fragment B (written $A \triangleright B$) if, typically, a human reading A would infer that B is most likely true (Dagan et al., 2005). Here, A is called the *premise* and B is called the *hypothesis*. For our purposes, we need a more formal definition of entailment. i.e. a definition in mathematical terms that matches the intuitive definition.

Entailment cannot be modeled as a material implication $A \Rightarrow B$ for two reasons: First, a material implication $A \Rightarrow B$ is true if B is true. Thus, "It rains" would entail "Paris is in France" – which is not the usual understanding of entailment. Propositional logic knows no satisfying way to avoid this. We could write $A \triangleright B := (A \Rightarrow B) \wedge (\neg A \Rightarrow \neg B)$; but that is just equivalent to $A \Leftrightarrow B$, which is not what entailment means. The second problem with defining entailment as a logical implication is that it does not allow for exceptions. For example, "I obtained a university diploma" entails "I have a university diploma", even if diplomas can be withdrawn in rare cases of fraud. Propositional logic has no means to say that an implication holds "usually" or "in the majority of cases".

Therefore, previous work (Glickman et al., 2005) has proposed a probabilistic definition of entailment. In what follows, we assume a probabilistic universe Ω and two events (the *premise* A and the *hypothesis* B). Glickman et al. (2005) then defines

Definition 3.1 (Entailment (Glickman et al., 2005))**.**

$$A \triangleright_G B := P(B|A) > P(B)$$

This definition says that A entails B if A increases the probability of B. Unfortunately, this definition has several problems: First, it is symmetric. We show in Proposition A.1 in the appendix that $(A \triangleright_G B) \Leftrightarrow (B \triangleright_G A)$. For example, "I live in Paris" $\triangleright_G$ "I live in France", because the probability of living in France increases to 100% once we know the person lives in Paris. However, knowing that someone lives in France also increases the probability that this person lives in Paris (from one in several million cities in the world to one in several thousand cities in France). Therefore "I live in France" $\triangleright_G$ "I live in Paris" – which is not our common understanding of entailment.

The second problem with Definition 3.1 is that $A \triangleright_G B$ even if A increases the probability of B only marginally. For example "I play in the lottery" $\triangleright_G$ "I win the lottery". This is because the probability of winning the lottery increases by playing in the lottery. Again, this is not our usual understanding of entailment.

Therefore, we propose to add the condition $P(B|A) > \theta$, where θ is a threshold for the acceptance of an entailment (say, 90%). Thus, our definition becomes $A \triangleright_\theta B := P(B|A) > P(B) \wedge P(B|A) > \theta$. This also makes the definition asymmetric, thus solving both the first problem and the second problem.

However, the definition is still vulnerable to a third problem: It may get carried away by hypotheses B with a high baseline probability. For example, most people survive the yearly Flu season. Washing your hands further decreases the risk of attracting the Flu (and thus increases the probability of survival). Hence "Alice washes her hands this Monday" $\triangleright_\theta$ "Alice survives this year's Flu season". This is because (1) washing hands indeed increases the probability of survival, and (2) the probability of surviving is already larger than θ (for $\theta = 90\%$). However, we would not say that the entailment holds. To guard against such cases, we propose to add another condition, $P(\neg A|\neg B) > \theta$. Our definition is thus:

Definition 3.2 (Entailment)**.**

$$\begin{aligned} A \triangleright B := & P(B|A) > P(B) \\ & \wedge P(B|A) > \theta \\ & \wedge P(\neg A|\neg B) > \theta \end{aligned}$$

with a given constant parameter $\theta \in [0; 1]$.

We write $A \ntriangleright B$ to say that A does not entail B. We can then use our notion of entailment to define contradiction and neutrality.

Definition 3.3 (Contradiction)**.**

$$A \blacktriangleright B := A \triangleright \neg B$$

Definition 3.4 (Neutrality)**.**

$$A \multimap B := A \ntriangleright B \wedge A \not\blacktriangleright B$$

3.2 Deriving New Instances

We can now use our definition of entailment to derive new premise-hypothesis pairs from a given pair. In what follows, let us denote the negation of a sentence A by $\neg A$. Formally, $\neg A := \Omega - A$. For example, the negation of "I live in Paris" is "I don't live in Paris". The negation of natural language sentences is a research topic on its own. For example, the negation of Noam Chomsky's famous nonsensical sentence "Colorless green ideas sleep furiously" is not "Colorless green ideas do not sleep furiously", as both are nonsensical. We refer the reader to Horn (1989); Löbner (2000); Penka (2015) and Homer et al. (2019) for a discussion. Here, we assume that both the premise and the hypothesis of a textual entailment instance are simple sentences that can be negated.

Now assume that we have $A \triangleright B$. Then Definition 3.2 allows us to formally derive $\neg B \triangleright \neg A$ (Proposition A.2 in the appendix). For example, "I live in Paris" $\triangleright$ "I live in France", and hence "I don't live in France" $\triangleright$ "I don't live in Paris". This type of reasoning is known as *Modus Tollens*. Table 1 shows other ways to derive new instances from a given instance, together with references to their proofs. A particularly interesting result is that $\blacktriangleright$ is symmetric, i.e., $(A \blacktriangleright B) \Leftrightarrow (B \blacktriangleright A)$.

Some of the derivations in Table 1 give us a label that an instance cannot have, rather than telling us which label it must have. We call such a label a *rejected label*. For example, an instance with the label $A \triangleright B$ (*entailment*) generates a new instance with the rejected label $\neg A \ntriangleright B$ (*non-entailment*, $\neg A$ does not entail B). This means that the true label cannot be an *entailment*, and that it has to be either *neutral* or a *contradiction*.

We are interested in entailments that logically follow from $A \triangleright B$, from $A \blacktriangleright B$, from $A \multimap B$ and from $A \ntriangleright B$, as these are the labels that common textual entailment datasets use: MNLI and SNLI use the first three labels, while RTE uses the first and last label. While Table 1 shows all derivations that must hold, Table 8 (in the appendix) shows all other hypothetical derivations, and proves them wrong. We can thus use Table 1 to derive, for a given labeled instance, new labeled instances. Most of these contain negation.

3.3 Unlikelihood Loss

The previous step has given us a way to derive new labeled instances – with either rejected or accepted labels. For the rejected labels, we want to penalize the likelihood of a language model predicting the rejected label. For this purpose, we use the *Unlikelihood Loss*. This loss has been used in many tasks, including in language modeling (Hosseini et al., 2021; Noji and Takamura, 2020) and text generation (Welleck et al., 2019). In our case, the loss is defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N} v_n log(p_{n,y_n}) + (1-v_n) log(1-p_{n,y_n})$$

Here, n runs over all N instances of the dataset. For each instance n and label y, $p_{n,y}$ is the score that the model assigns to the label y for the instance n. To each n we associate a ground truth label y_n, and we know whether this label is accepted or rejected. To distinguish these two cases, v_n is

Original	Derivation	Proof	Example
$A \rhd B$	$A \rhd B$	-	I live in Paris $\rhd$ I live in France
	$A \blacktriangleright \neg B$	Per definition of $\blacktriangleright$	I live in Paris $\blacktriangleright$ I don't live in France
	$\neg B \rhd \neg A$	Proposition A.2 (Modus Tollens)	I don't live in France $\rhd$ I don't live in Paris
	$\neg A \ntriangleright B$	Proposition A.3	I don't live in Paris $\ntriangleright$ I don't live in France
	$\neg B \blacktriangleright A$	Per definition of $\blacktriangleright$ with Modus Tollens	I don't live in France $\blacktriangleright$ I live in Paris
	$A \ntriangleright \neg B$	Proposition A.4	I live in Paris $\ntriangleright$ I don't live in France
	$B \ntriangleright \neg A$	Proposition A.5	I live in France $\ntriangleright$ I don't live in Paris
	$\neg B \ntriangleright A$	Proposition A.6	I don't live in France $\ntriangleright$ I live in Paris
$A \blacktriangleright B$	$A \blacktriangleright B$	-	I live in Paris $\blacktriangleright$ I live in Italy
	$A \rhd \neg B$	Per definition of $\blacktriangleright$	I live in Paris $\rhd$ I don't live in Italy
	$\neg A \not\blacktriangleright B$	Proposition A.7	I don't live in Paris $\not\blacktriangleright$ I live in Italy
	$B \rhd \neg A$	Equivalent to Proposition A.2 by definition	I live in Italy $\rhd$ I don't live in Paris
	$B \blacktriangleright A$	Per definition of $\blacktriangleright$	I live in Italy $\blacktriangleright$ I live in Paris
	$B \not\blacktriangleright \neg A$	Reduces to $A \rhd B' \Rightarrow \neg B' \ntriangleright A$ with $B' = \neg B$	I live in Italy $\not\blacktriangleright$ I don't live in Paris
	$\neg B \not\blacktriangleright A$	Apply Proposition A.2 then A.3	I don't live in Italy $\not\blacktriangleright$ I live in Paris
	$A \ntriangleright B$	Reduces to $A \rhd B' \Rightarrow A \ntriangleright \neg B'$ with $B' = \neg B$	I live in Paris $\ntriangleright$ I live in Italy
$A \multimap B$	$A \multimap B$	-	I live in France $\multimap$ I live in Paris
	$A \multimap \neg B$	Proposition A.8	I live in France $\multimap$ I don't live in Paris
$A \ntriangleright B$	$A \ntriangleright B$	-	I live in France $\ntriangleright$ I live in Paris
	$\neg B \ntriangleright \neg A$	Proposition A.9	I don't live in Paris $\ntriangleright$ I don't live in France

Table 1: Rules for deriving textual entailment instances. The propositions and their proofs are in Appendix A.

an indicator that takes the value 1 if the label is accepted, and the value 0 if the label is rejected. Our loss is thus the sum of the cross-entropy loss of the accepted labels and the unlikelihood loss of the rejected labels.

3.4 Dataset Augmentation

To augment a textual entailment dataset with negated instances, we consider all instances one by one. We first check if the instance consists of a grammatically correct single-sentence premise and single-sentence hypothesis. We use DistillBERT (Sanh et al., 2019) to that end, a model that was finetuned on the The Corpus of Linguistic Acceptability (COLA) dataset (Warstadt et al., 2019). If the instance does not pass this test, we skip it. Otherwise, we check if we can negate both the premise and the hypothesis of the instance. We use the method developed by Hosseini et al. (2021) for this purpose, a rule-based approach with pre-defined rules written in Semgrex (Chambers et al., 2007). It takes as input a sentence with part-of-speech tags (POS tags), the dependency parse, and the morphological features of the words, and it produces as output a negated sentence. We used Stanza (Qi et al., 2020) to get the POS tags, the dependency parse, and the morphological features. Here is an example: "The man is somewhere near the parade" $\rightsquigarrow$ "The man is **nowhere** near the parade" .

If both the premise and the hypothesis can be negated, we derive possible new instances as per Table 1. We illustrate this data augmentation process with an instance from SNLI[2]:

Premise: The two boys are in martial arts poses in an outside basketball court.
Hypothesis: The two boys are outdoors.
Label: $A \rhd B$ (Entailment)

Derivation Example $A \blacktriangleright \neg B$

Premise: The two boys are in martial arts poses in an outside basketball court.
Hypothesis: The two boys are **not** outdoors.
Label: Contradiction

Derivation Example $\neg B \rhd \neg A$

Premise: The two boys are **not** outdoors.
Hypothesis: The two boys are **not** in martial arts poses in an outside basketball court.
Label: Entailment

[2] Since SNLI instances are always about a given scene, we added the determiner "the" here.

Derivation Example $\neg B \blacktriangleright A$

Premise: The two boys are **not** outdoors.
Hypothesis: The two boys are in martial arts poses in an outside basketball court.
Label: Contradiction

Derivation Example $\neg A \not\triangleright B$

Premise: The two boys are **not** in martial arts poses in an outside basketball court.
Hypothesis: The two boys are outdoors.
Label: Not Entailment

This last example should actually be labeled *neutral*, as the boys can be outside without martial arts poses. However, not all pairs of $\neg A$ and B are neutral when $A \triangleright B$, they can also be in a contradiction: with A="I live in Paris" and B="I live in the capital of France", we have $A \triangleright B$, and $\neg A \blacktriangleright B$. The relation of $\neg A$ and B thus cannot be determined just by knowing $A \triangleright B$. However, our approach can still generate a rejected label that can be used for training.

4 Experiments

We conducted several experiments to investigate the robustness of models trained with our data augmentation technique, TINA, for the task of textual entailment with negation.

4.1 Settings

Datasets. We use the most common datasets for textual entailment, namely Stanford Natural Language Inference (SNLI) (Bowman et al., 2015), Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2018), and Pascal RTE (RTE) (Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007, 2008; Bentivogli et al., 2009). Each dataset comes with a *Train* dataset for training, and a *Dev* dataset for development. Following Hossain et al. (2020), we used the development dataset as the testing set because the GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) benchmarks do not provide gold labels for the test splits.

In addition, each dataset also has a negated variant *Neg*, created by Hossain et al. (2020) to evaluate the understanding of negation in language models. Each negated benchmark was created by randomly selecting 500 premise-hypothesis pairs from the datasets of SNLI, MNLI, and RTE. For each instance, 3 new pairs were generated by adding the negation "not", as follows:

- Adding a negation to the premise and keeping the original hypothesis
- Adding a negation just to the hypothesis and keeping the original premise
- Adding a negation to the premise and the hypothesis

Finally, for each dataset, we generate an augmented variant *Aug* by our methodology from Section 3. We made sure that the generated instances are not in the negated benchmarks. Table 2 shows the number of instances from the training set of each dataset that were negated before deriving new instances. Table 3 shows the sizes of the datasets.

Dataset	*Train*	*Negated Train*
SNLI	550,152	78,116
MNLI	392,702	199,648
RTE	2,490	2,308

Table 2: Number of instances in each training dataset that were negated

Dataset	*Train*	*Dev*	*Aug*	*Neg*
SNLI	550,152	10,000	233,024	1,500
MNLI	392,702	9,815	601,441	1,500
RTE	2,490	277	2,408	1,500

Table 3: Number of instances in each dataset

Models. We want to see whether TINA makes transformer-based models more robust to negation in textual entailment. Our experiments cover the following models:

BERT (Devlin et al., 2019) is a pretrained language model that consists of an encoder block of a stack of transformer layers. It was pretrained on two tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In our experiments we use BERT-Base Cased with 110M parameters.

RoBERTa (Liu et al., 2019) is a pretrained model that has a similar architecture to BERT but achieves better performance on many NLP tasks. In contrast to BERT, it was pretrained longer with bigger batches on a larger dataset, and only for the MLM task, by dynamically changing the masked tokens after each training epoch. In our experiments, we used RoBERTa-Base with 123M parameters.

XLNet (Yang et al., 2019) is a transformer-based model that was pretrained on a task called Permutation Language Modeling (PLM). PLM is the task of capturing bidirectional context by training a model on all possible permutation of words in a sentence. In our experiments, we used XLNet-Base Cased with 110M parameters.

BART (Lewis et al., 2020) is a sequence-to-sequence model composed of an encoder block like BERT and a decoder block like GPT. The pretrained task consists of reconstructing a corrupted text to its original after applying different noising functions such as token masking, token infilling, and sentence permutations. In our experiments, we used BART-base with 139M parameters.

GPT-2 (Radford et al., 2019) is a model consisting of a decoder block of transformers layers. It was pretrained to predict the next word given all the previous words in a sentence. GPT-2 has 1.5 billion parameters.

We finetune BERT (Base Cased), RoBERTa (Base), and XLNet (Base Cased) on each training set and evaluate them on each testing set. We use the same hyperparameters as Hossain et al. (2020) for the number of epochs, batch size, learning rate, and weight decay. We recall them in Table 6. However, unlike the original work, we set the maximum sequence length to 512 instead of 128. We also applied our approach to BART (Base) and GPT-2. We split the training dataset as 90/10 for training and validation sets for these two models. We evaluated on each testing set with the best-performing models based on the validation set. We carried out a basic hyperparameter search and describe the hyperparameters that we found in Table 7. All models were trained on an NVIDIA A100 GPU with 40GB memory.

Competitors. The only other approach that specifically targets negation in textual entailment is BERTNOT (Hosseini et al., 2021). It was trained to model negation in the MLM task, and then it was finetuned on each TE training set. For reference, we also show the performance of a T5-Base model. This model is very powerful, as it was pretrained on a mixture of NLP tasks that include textual entailment, coreference resolution, linguistic acceptability, and semantic equivalence.

4.2 Results

Table 4 shows the performance of TINA applied to different transformer-based models averaged over 3 runs. TINA$^-$ is a variant of TINA that does not generate instances with rejected labels. We show, for each model, how the performance changes when TINA$^-$ and TINA are used. We compute a binomial confidence interval for each result (at a confidence level of $\alpha = 0.05$), based on the total number of instances and the number of correctly predicted labels.

The main outcome is that, on the negated datasets, TINA$^-$ always improves the results, and TINA improves the results even more. At the same time, the augmentation techniques do not lower the results significantly on the original datasets. This is true across all models.

On the SNLI dataset, the improvement of the performance is considerable, with gains up to 20 percentage points, depending on the model. On MNLI, the gains are less. We assume that this is because MNLI contains many ungrammatical sentences, and also because it already contains some proportion of negated training examples. Nevertheless, the gains of TINA are still significant. On RTE, TINA and TINA$^-$ are identical, as the dataset only has two labels (*entailment* and *non-entailment*). The confidence intervals on RTE_{Dev} are much larger, because the dataset is much smaller. Nevertheless, the gains on the negated dataset are significant, and can reach up to 21 percentage points, depending on the model.

For reference, we also show the performance of an off-the-shelf pretrained T5-Base model. It has a very good performance, and most notably outperforms our competitor BERTNOT significantly on the negated datasets. We assume that this is because it was pretrained on a large mixture of NLP tasks. Nevertheless, our method comes close to T5 on RTE, and outperforms the T5 model on SNLI and MNLI.

Most importantly, however, our approach serves its purpose, in that it increases the performance of transformer-based models on negated textual entailment by a large margin, across different models and all datasets. With this, our approach improves

Model	*SNLI*		*MNLI*		*RTE*	
	$SNLI_{Dev.}$	$SNLI_{Neg.}$	$MNLI_{Dev.}$	$MNLI_{Neg.}$	$RTE_{Dev.}$	$RTE_{Neg.}$
BERTNOT (Hosseini et al., 2021)	89.00$_{\pm 0.62}$	45.96$_{\pm 2.53}$	84.31$_{\pm 0.73}$	60.89$_{\pm 2.51}$	69.68$_{\pm 5.65}$	74.47$_{\pm 2.27}$
Pretrained T5-Base (Beam Search)	78.61$_{\pm 0.81}$	60.33$_{\pm 2.56}$	86.04$_{\pm 0.70}$	66.46$_{\pm 2.48}$	66.06$_{\pm 6.12}$	83.13$_{\pm 2.04}$
Pretrained T5-Base (Greedy Search)	78.29$_{\pm 0.81}$	61.40$_{\pm 2.48}$	85.61$_{\pm 0.71}$	67.00$_{\pm 2.42}$	67.87$_{\pm 6.08}$	82.60$_{\pm 2.00}$
BERT	89.19$_{\pm 0.62}$	49.10$_{\pm 2.55}$	83.38$_{\pm 0.75}$	65.21$_{\pm 2.45}$	67.62$_{\pm 5.83}$	58.30$_{\pm 2.54}$
+ TINA$^-$	+ 0.0	+ 3.56	- 1.33	+ 4.29	+ 0.84	+ 21.63
+ TINA	- 0.21	+ 20.09	- 2.81	+ 4.21	-	-
RoBERTa	90.18$_{\pm 0.59}$	54.46$_{\pm 2.58}$	86.55$_{\pm 0.69}$	66.93$_{\pm 2.48}$	76.54$_{\pm 5.33}$	74.35$_{\pm 2.28}$
+ TINA$^-$	- 0.1	+ 0.89	- 0.45	+ 1.62	+ 0.11	+ 7.18
+ TINA	- 0.05	+ 13.05	- 0.45	+ 2.04	-	-
XLNet	89.98$_{\pm 0.60}$	53.77$_{\pm 2.56}$	85.76$_{\pm 0.70}$	67.06$_{\pm 2.48}$	70.15$_{\pm 5.75}$	68.08$_{\pm 2.41}$
+ TINA$^-$	- 0.26	+ 2.31	- 0.31	+ 3.03	- 5.66	+ 6.65
+ TINA	- 0.34	+ 12.8	- 1.01	+ 3.8	-	-
BART	89.79$_{\pm 0.60}$	53.17$_{\pm 2.56}$	84.90$_{\pm 0.73}$	66.60$_{\pm 2.49}$	70.51$_{\pm 5.73}$	60.30$_{\pm 2.53}$
+ TINA$^-$	- 0.20	- 0.6	- 0.65	+ 3.04	+ 0.10	+ 17.03
+ TINA	- 0.09	+ 17.6	- 1.37	+ 3.66	-	-
GPT-2	87.56$_{\pm 0.66}$	48.77$_{\pm 2.55}$	80.94$_{\pm 0.79}$	62.24$_{\pm 2.52}$	61.97$_{\pm 6.08}$	57.37$_{\pm 2.55}$
+ TINA$^-$	+ 0.04	+ 2.09	- 0.37	+ 4.73	+ 4.45	+ 17.56
+ TINA	+ 0.01	+ 6.67	- 0.42	+ 5.93	-	-

Table 4: Results of our approach applied to different language models on different textual-entailment datasets. Accuracies are averaged across 3 runs. Significant changes have a gray background.

over the current state of the art (Hosseini et al., 2021).

4.3 Qualitative Analysis

To better understand the performances of TINA, we manually checked a sample of sentences from each augmented dataset. For SNLI, we find that the sentences are simple. They just contain one verb, which is easy for Hosseini et al. (2021)'s tool to negate. In contrast, MNLI and RTE have longer and more complex premises, which are not always grammatical. This leads to problematic cases where the negation does not work, which we group into the following categories:

Ungrammatical sentences cannot be negated properly: "would i swim that river every night twice if that's what it took you know i don't care whatever it would take i have real sympathy for those people i really do and you can." ⇝ "would **not** i swim that river every night twice if that 's what it took you know i don't care whatever it would take i have real sympathy for those people i really do and you can."

Conjunctions are negated only in their first conjunct: "The motion set waves of nausea running through him, but he could see the doctor" ⇝ "The motion **did not** set waves of nausea running through him, but he could see the doctor" . The same goes for adjectives and prepositions that take a role akin to a conjunction, as in "despite concerns about the drinking water".

Verbs of assertion are negated, but not the assertion itself: "The actor was outside a movie theater in central London's Leicester Square, London's Metropolitan Police said" ⇝ "The actor was outside a movie theater in central London's Leicester Square, London's Metropolitan Police **did not** say". In this case, the negation does not work as intended, as the main verb merely states the source of the assertion. In other cases, the main verb may indeed be the intended target of the negation.

Negation errors occur at times with Hosseini et al. (2021)'s tool, as e.g. in "cannot not do" and "has did not given".

Our filtering step with DistillBERT (Sanh et al., 2019) was apparently insufficient to remove the ungrammatical sentences. For the conjuncts, we found that the erroneous negation is mostly harmless: if a conjunction is negated only in its first conjunct, that might still be the conjunct that is relevant for the entailment. The same goes for verbs of assertion: the entailment may sometimes target the fact of asserting something (in which case the negation works correctly). Negation errors, too, may be harmless: while these can disturb a human reader, they may still yield useful signals for a machine learning model.

The negation of sentences thus remains a challenge in practice. It is, however, largely orthogonal to our contribution of creating negated training examples for textual entailment. We are thus hopeful that an improvement of these tools will confer even higher performance gains to TINA.

5 Conclusion

In this paper, we have studied the problem of negation in textual entailment in detail. We have argued that the previous formal definition of textual entailment is problematic, and we have proposed a new probabilistic definition. Based on this definition, we have proposed TINA, a principled negated data augmentation technique. TINA can be combined with the unlikelihood loss to improve the robustness of language models to negation in textual entailment tasks. Our experimental results across different negated textual entailment benchmarks show that our method can significantly increase the performance of different transformer-based models. Future work can explore how different loss functions, such as contrastive loss, could be used with our augmented datasets.

Acknowledgements. This work was partially funded by ANR-20-CHIA-0012-01 ("NoRDF").

Limitations

One limitation of our approach is that it presupposes premise-hypothesis pairs that consist of simple, negatable sentences. We already filter out sentences that do not conform, but many cases of incorrect negations remain (Section 4.3). The correct negation of sentences thus remains an open challenge.

Our probabilistic definition of entailment can also be further scrutinized. While we believe that it filters out most counter-intuitive entailments, it may still be possible to come up with counter-intuitive examples that fulfill our definition. It is even possible that this cannot be avoided at all, as the textual entailment task itself suffers from a degree of vagueness.

Finally, our method focuses purely on the generation of training instances. However, it may be possible that specified models (one for negated instances and one for affirmative instances) lead to better results.

References

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Conference on Empirical Methods in Natural Language Processing*.

Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine De Marneffe, Daniel Ramage, Eric Yeh, and Christopher D Manning. 2007. Learning alignments and leveraging natural logic. In *ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*.

Danilo Giampiccolo, Hoa Trang Dang, Bernardo Magnini, Ido Dagan, Elena Cabrio, and Bill Dolan. 2008. The fourth pascal recognizing textual entailment challenge. In *TAC*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *ACL-PASCAL workshop on textual entailment and paraphrasing*.

Oren Glickman, Ido Dagan, and Moshe Koppel. 2005. Web based probabilistic textual entailment.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *The Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Bharath Hariharan and Ross Girshick. 2017. Low-shot visual recognition by shrinking and hallucinating features. In *IEEE International Conference on Computer Vision*.

Chadi Helwe, Chloé Clavel, and Fabian M Suchanek. 2021. Reasoning with transformer-based models: Deep learning, but shallow reasoning. In *Conference on Automated Knowledge Base Construction*.

Vincent Homer, Lisa Matthewson, Cécile Meier, Hotze Rullman, and Thomas Ede Zimmermann. 2019. Negative polarity. *Blackwell companion to semantics, Wiley (forthcoming)*.

Laurence R Horn. 1989. A natural history of negation.

Md Mosharaf Hossain, Dhivya Chinnappa, and Eduardo Blanco. 2022. An analysis of negation in natural language understanding corpora. In *Annual Meeting of the Association for Computational Linguistics*.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. An analysis of natural language inference benchmarks through the lens of negation. In *Conference on Empirical Methods in Natural Language Processing*.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Annual Meeting of the Association for Computational Linguistics*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sebastian Löbner. 2000. Polarity in natural language: Predication, quantification and negation in particular and characterizing sentences. *Linguistics and Philosophy*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Annual Meeting of the Association for Computational Linguistics*.

Hiroshi Noji and Hiroya Takamura. 2020. An analysis of the utility of explicit negative examples to improve the syntactic abilities of neural language models. *arXiv preprint arXiv:2004.02451*.

Magdalini Paschali, Walter Simson, Abhijit Guha Roy, Muhammad Ferjad Naeem, Rüdiger Göbl, Christian Wachinger, and Nassir Navab. 2019. Data augmentation with manifold exploring geometric transformations for increased performance and robustness. *arXiv preprint arXiv:1901.04420*.

Doris Penka. 2015. Negation and polarity. In *The Routledge handbook of semantics*.

Adam Poliak. 2020. A survey on recognizing textual entailment as an NLP evaluation. In *First Workshop on Evaluation and Comparison of NLP Systems*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In *Conference on Empirical Methods in Natural Language Processing*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex Bronstein. 2018. Delta-encoder: an effective sample synthesis method for few-shot object recognition. *Advances in Neural Information Processing Systems*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner. *arXiv preprint arXiv:2104.14690*.

Siyuan Wang, Wanjun Zhong, Duyu Tang, Zhongyu Wei, Zhihao Fan, Daxin Jiang, Ming Zhou, and Nan Duan. 2022. Logic-driven context extension and data augmentation for logical reasoning of text. In *Findings of the Association for Computational Linguistics*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*.

A Proofs

Proposition A.1. For all events A and B, if $A \rhd_G B$ then $B \rhd_G A$ (with $\rhd_G$ defined in Definition 3.1).

Proof. Let A and B be two events with $A \rhd_G B$. We have:

$$\begin{aligned} P(B|A) &= \frac{P(A\cap B)}{P(A)} > P(B) \\ \Rightarrow P(A|B) &= \frac{P(A\cap B)}{P(B)} > P(A) \\ \Rightarrow B \rhd_G A \end{aligned}$$

$\square$

Proposition A.2 (Modus Tollens)**.** For all events A and B, if $A \rhd B$ then $\neg B \rhd \neg A$.

Proof. By definition of $A \rhd B$ we have all the followings:
- $P(B|A) > P(B)$
- $P(B|A) > \theta$
- $P(\neg A|\neg B) > \theta$

We need to prove all the followings:
- $P(\neg A|\neg B) > P(\neg A)$
- $P(\neg A|\neg B) > \theta$
- $P(\neg\neg B|\neg\neg A) > \theta$

The last condition is equivalent to $P(B|A) > \theta$. Hence we need to prove only $P(\neg A|\neg B) > P(\neg A)$.

To simplify the proof we introduce: $a = P(A \cap \neg B)$ $b = P(\neg A \cap B)$ $c = P(A \cap B)$ $d = P(\neg A \cap \neg B)$ (summarized in Table 5). Then we have:

$$\begin{aligned} &P(B|A) = \frac{P(A\cap B)}{P(A)} > P(B) \\ &\Rightarrow \frac{P(A\cap B)}{P(A\cap B)+P(A\cap\neg B)} > P(A\cap B)+P(\neg A\cap B) \\ &\Rightarrow \frac{c}{a+c} > b+c \\ &\Rightarrow \frac{1-a-b-d}{1-b-d} > 1-a-d \\ &\Rightarrow \frac{d}{a+d} > b+d \\ &\Rightarrow P(\neg A|\neg B) > P(\neg A) \end{aligned}$$

$\square$

Proposition A.3. For all events A and B, if $A \rhd B$ then $\neg A \not\rhd B$.

Proof. Assume that there exist A and B such that $A \rhd B$ and $\neg A \rhd B$. Then $P(B|A) > P(B)$ and $P(B|\neg A) > P(B)$. Hence, we have:

$$\begin{aligned} &P(B) = P(A) \times P(B|A) + P(\neg A) \times P(B|\neg A) \\ &\Rightarrow P(B) > P(A) \times P(B) + P(\neg A) \times P(B) \\ &\Rightarrow P(B) > P(B) \end{aligned}$$

This is a contradiction, which proves the claim. $\square$

Proposition A.4. For all events A and B, if $A \rhd B$ then $A \not\rhd \neg B$.

Proof. Assume $A \rhd B$ and $A \rhd \neg B$. We have $P(B|A) > P(B)$ and $P(B|\neg A) > P(B)$. Hence $P(B) > P(B)$. Contradiction. $\square$

Proposition A.5. For all events A and B, if $A \rhd B$ then $B \not\rhd \neg A$.

Proof. If $B \rhd \neg A$ then by Modus Tollens (Proposition A.2), $A \rhd \neg B$. By proposition A.4 we have $A \not\rhd \neg B$. Contradiction. $\square$

Proposition A.6. For all events A and B, if $A \rhd B$ then $\neg B \not\rhd A$.

Proof. If $\neg B \rhd A$ then by Modus Tollens (Proposition A.2), $\neg A \rhd B$. By proposition A.3 we have $\neg A \not\rhd B$. Contradiction. $\square$

Proposition A.7. For all events A and B, if $A \blacktriangleright B$ then $\neg A \not\blacktriangleright B$.

Proof. By definition, our proposition is equivalent to $(A \rhd \neg B) \Rightarrow (\neg A \not\rhd \neg B)$. This is true according to Proposition A.3. $\square$

Proposition A.8. For all events A and B, if $A \multimap B$ then $A \multimap \neg B$.

Proof.

$$\begin{aligned} A \multimap B &\equiv (A \not\rhd B \text{ and } A \not\rhd \neg B) \\ &\equiv A \not\rhd \neg B \text{ and } A \not\rhd \neg\neg B \\ &\equiv A \multimap \neg B \end{aligned}$$

$\square$

Proposition A.9. For all events A and B, if $A \not\rhd B$ then $\neg B \not\rhd \neg A$.

Proof. Assume A and B such that $A \not\rhd B$ and $\neg B \rhd \neg A$. Then by Modus Tollens (Proposition A.2), $\neg\neg A \rhd \neg\neg B$, which we can restate as $A \rhd B$. Contradiction. $\square$

	B	$\neg B$
A	c	a
$\neg A$	b	d

Table 5: Shorthand notations. For example, b is equal to $P(\neg A \cap B)$.

B Hyperparameters

Tables 6 and 7 show the hyperparameters that we used in our experiments (Section 4).

	SNLI			*MNLI*			*RTE*		
	BERT	*RoBERTa*	*XLNet*	*BERT*	*RoBERTa*	*XLNet*	*BERT*	*RoBERTa*	*XLNet*
Epochs	3	3	3	3	3	3	50	10	50
Batch Size	32	32	32	32	32	32	8	16	8
Learning Rate	1e-5	1e-5	1e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5
Weight Decay	0.1	0.1	0.1	0	0	0	0	0	0

Table 6: Hossain et al. (2020) hyperparameter configurations

	SNLI		*MNLI*		*RTE*	
	BART	*GPT-2*	*BART*	*GPT-2*	*BART*	*GPT-2*
Epochs	10	10	10	10	10	10
Batch Size	32	32	32	32	8	8
Learning Rate	1e-5	1e-5	2e-5	2e-5	2e-5	2e-5
Weight Decay	0.1	0.1	0	0	0	0

Table 7: BART and GPT-2 hyperparameter configurations

C Figures

Figure 1 shows a graphical illustration of the performances in Table 4.

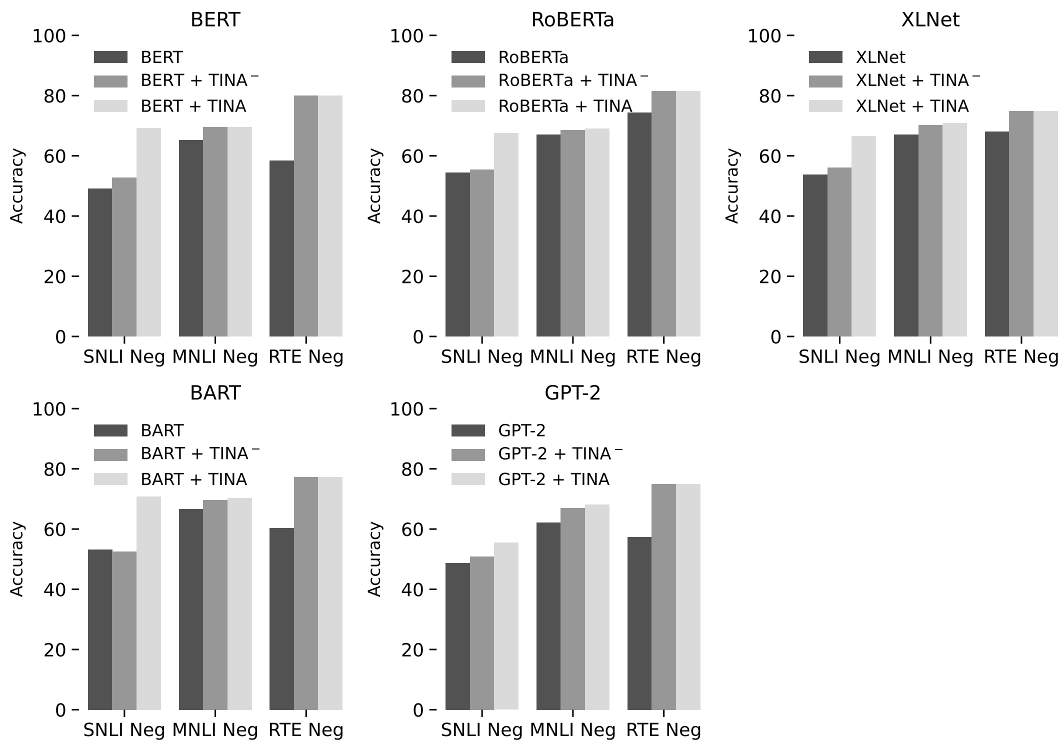

Figure 1: Evaluation of different finetuning methods applied to different transformer-based models on the negated textual entailment datasets. Accuracies are averaged across 3 runs.

D Table of Derivations

Table 8 presents all derivations that are not in Table 1. We show for each of them that they do not hold. This is done either by a counterexample, by reducing them to another derivation that does not hold, or by showing they contradict other true derivations. As before, we use the notations from Table 5.

Original	Derivation	Counterexample, reduction, or proof	Illustrative counterexample
$A \rhd B$	$A \not\rhd B$	Trivial	I live in Paris $\rhd$ I live in France
	$A \rhd \neg B$	$a = 0, b = 0, c = 0.125, d = 0.875, \theta = 0$	I live in Paris $\not\rhd$ I don't live in France
	$\neg A \rhd B$	$a = 0, b = 0, c = 0.125, d = 0.875, \theta = 0$	I don't live in Paris $\not\rhd$ I live in France
	$\neg A \rhd \neg B$	$a = 0.02, b = 0.72, c = 0.18, d = 0.08, \theta = 0$	I don't live in Paris $\not\rhd$ I don't live in France
	$B \rhd A$	$a = 0, b = 0.01, c = 0.01, d = 0.98, \theta = 0.8$	I live in France $\not\rhd$ I live in Paris
	$B \rhd \neg A$	Contradicts propositions A.3 and A.2 (Modus Tollens)	I live in France $\not\rhd$ I don't live in Paris
	$\neg B \rhd A$	$a = 0, b = 0, c = 0.01, d = 0.99, \theta = 0$	I don't live in France $\not\rhd$ I live in Paris
	$\neg A \not\rhd \neg B$	$a = 0, b = 0, c = 0.01, d = 0.99, \theta = 0$	I don't live in France $\rhd$ I don't live in France
	$B \not\rhd A$	$a = 0, b = 0, c = 0.01, d = 0.99, \theta = 0$	I live in France $\rhd$ I live in France
$A \blacktriangleright B$	$A \not\blacktriangleright B$	Trivial	I live in Paris $\blacktriangleright$ I live in Italy
	$A \blacktriangleright \neg B$	Reduces to $A \rhd B' \Rightarrow A \rhd \neg B'$ with $B' = \neg B$	I live in Paris $\not\blacktriangleright$ I don't live in Italy
	$\neg A \blacktriangleright B$	Reduces to $A \rhd B' \Rightarrow \neg A \rhd B'$ with $B' = \neg B$	I don't live in Paris $\not\blacktriangleright$ I live in Italy
	$\neg A \blacktriangleright \neg B$	Reduces to $A \rhd B' \Rightarrow \neg A \rhd \neg B'$ with $B' = \neg B$	I don't live in Paris $\not\blacktriangleright$ I don't live in Italy
	$B \blacktriangleright \neg A$	Reduces to $A \rhd B' \Rightarrow \neg B' \rhd A$ with $B' = \neg B$	I live in Italy $\not\blacktriangleright$ I don't live in Paris
	$\neg B \blacktriangleright A$	Reduces to $A \rhd B' \Rightarrow B' \rhd \neg A$ with $B' = \neg B$	I don't live in Italy $\not\blacktriangleright$ I live in Paris
	$\neg B \blacktriangleright \neg A$	Reduces to $A \rhd B' \Rightarrow B' \rhd A$ with $B' = \neg B$	I don't live in Italy $\not\blacktriangleright$ I don't live in Paris
	$\neg A \not\blacktriangleright \neg B$	Reduces to $A \rhd B' \Rightarrow \neg A \not\rhd \neg B'$ with $B' = \neg B$	I don't live in Paris $\blacktriangleright$ I live in Paris
	$B \not\blacktriangleright A$	Contradicts Proposition A.2 (Modus Tollens)	I live in Italy $\blacktriangleright$ I live In Paris
	$\neg B \not\blacktriangleright \neg A$	Reduces to $A \rhd B' \Rightarrow B' \not\rhd A$ with $B' = \neg B$	I don't live in Paris $\blacktriangleright$ I live in Paris
$A \multimap B$	$\neg A \multimap B$	$a = 0.02, b = 0.69, c = 0.06, d = 0.23, \theta = 0$	I don't live in France $\not\multimap$ I live In Paris
	$\neg A \multimap \neg B$	$a = 0.02, b = 069, c = 0.06, d = 0.23, \theta = 0$	I don't live in France $\not\multimap$ I don't live in Paris
	$B \multimap A$	$a = 0.02, b = 0.72, c = 0.08, d = 0.18, \theta = 0$	I live in Paris $\not\multimap$ I live in France
	$B \multimap \neg A$	$a = 0.02, b = 0.72, c = 0.08, d = 0.18, \theta = 0$	I live in Paris $\not\multimap$ I don't live in France
	$\neg B \multimap A$	$a = 0.02, b = 0.69, c = 0.06, d = 0.23, \theta = 0$	I win the lottery $\not\multimap$ I play the lottery
	$\neg B \multimap \neg A$	$a = 0.02, b = 0.69, c = 0.06, d = 0.23, \theta = 0$	I don't live in France $\not\multimap$ I live in Paris
$A \not\rhd B$	$A \not\rhd \neg B$	$a = 0.01, b = 0.01, c = 0, d = 0.98, \theta = 0$	I live in Paris $\rhd$ I don't live in Italy
	$\neg A \not\rhd B$	$a = 0.01, b = 0.01, c = 0, d = 0.98, \theta = 0$	I don't live in France $\rhd$ I don't live in Paris
	$\neg A \not\rhd \neg B$	$a = 0.02, b = 0.69, c = 0.06, d = 0.23, \theta = 0$	I live in France $\rhd$ I don't live in Paris
	$B \not\rhd A$	$a = 0.01, b = 0, c = 0.01, d = 0.98, \theta = 0.8$	I live in Paris $\rhd$ I live in France
	$B \not\rhd \neg A$	$a = 0.01, b = 0.01, c = 0, d = 0.98, \theta = 0$	I live in Paris $\rhd$ I don't live in France
	$\neg B \not\rhd A$	$a = 0.01, b = 0.01, c = 0, d = 0.98, \theta = 0$	I don't live in France $\rhd$ I don't live in Paris

Table 8: False derivations