AdvERSEM: Adversarial Robustness Testing and Training of LLM-based Groundedness Evaluators via Semantic Structure Manipulation

Kaustubh D. Dhole, Ramraj Chandradevan, Eugene Agichtein
Department of Computer Science
Emory University
{kaustubh.dhole, eugene.agichtein}@emory.edu

Abstract

Evaluating outputs from large language models (LLMs) presents significant challenges, especially as hallucinations and adversarial manipulations are often difficult to detect. Existing evaluation methods lack robustness against subtle yet intentional linguistic alterations, necessitating novel techniques for reliably assessing model-generated content. Training accurate and robust groundedness evaluators is key for mitigating hallucinations and ensuring the alignment of model or human-generated claims to real-world evidence. However, as we show, many models, while optimizing for accuracy, lack robustness to subtle variations of claims, making them unsuitable and brittle in real-world settings where adversaries employ purposeful and deceitful tactics like hedging to deceive readers, which go beyond surface-level variations. To address this problem, we propose **AdvERSEM**, a controllable adversarial approach to manipulating LLM output via Abstract Meaning Representations (AMR) to generate attack claims of multiple fine-grained types, followed by automatic verification of the correct label. By systematically manipulating a unique linguistic facet **AdvERSEM** provides an interpretable testbed for gauging robustness as well as useful training data. We demonstrate that utilizing these AMR manipulations during training across multiple fact verification datasets helps improve the accuracy and robustness of groundedness evaluation while also minimizing the requirement of costly annotated data. To encourage further systematic evaluation, we release **AdvERSEM-Test**, a manually verified groundedness test-bed.[1]

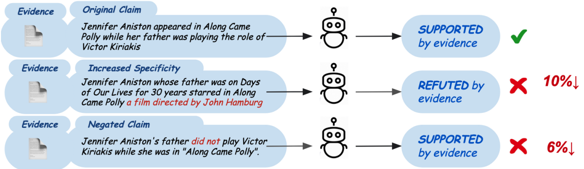

Evidence
Original Claim
Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis
SUPPORTED by evidence
✓
Evidence
Increased Specificity
Jennifer Aniston whose father was on Days of Our Lives for 30 years starred in Along Came Polly a film directed by John Hamburg
REFUTED by evidence
✗
10%↓
Evidence
Negated Claim
Jennifer Aniston's father did not play Victor Kiriakis while she was in "Along Came Polly".
SUPPORTED by evidence
✗
6%↓

Figure 1: Groundedness Evaluators falter when the claim is made more specific or is negated

1 Introduction

Evaluating the reliability of human or model-generated claims typically involves human judgment, which can be costly, and insufficiently sensitive to subtle manipulations in generated text. Automatic evaluators like LLM-Judges offer scalable alternatives for assessing critical properties like groundedness, and factual accuracy, yet these evaluators themselves often rely on LLMs and thus inherit their limitations, including susceptibility to hallucinations, adversarial perturbations and the transformers' non-compositional nature of training (Nandi et al., 2025). Developing robust automatic evaluators is particularly challenging, as subtle semantic changes (Lee et al., 2025; Raina et al., 2024), purposeful negations, or intentional manipulations like hedging (Paige et al., 2024) can cause dramatic degradation in evaluation reliability. Besides, most evaluation testbeds rarely offer fine-grained performance assessments, providing almost zero feedback to evaluation assessors. Hence, to improve trustworthiness, it is vital that groundedness are robust as well as provide fine-grained feedback of performance.

Groundedness evaluation or fact verification refers to the alignment of human or model-generated claims with real-world evidence. This alignment is crucial for assessing the truthfulness of statements against established knowledge. However, current language models often exhibit vulnerabilities that compromise their groundedness. For instance, the Llama-3.1-7B model fails to consistently predict the factual correctness of claims when claims are slightly altered to make them more specific or when they are simply negated as shown in Figure 1. In this work, we focus on improving groundedness evaluation by fine-tuning on adversarially generated examples. Specifically, using popular fact verification datasets, we construct ad-

[1] https://github.com/emory-irlab/adversem

*Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 395–408
November 8-9, 2025 ©2025 Association for Computational Linguistics

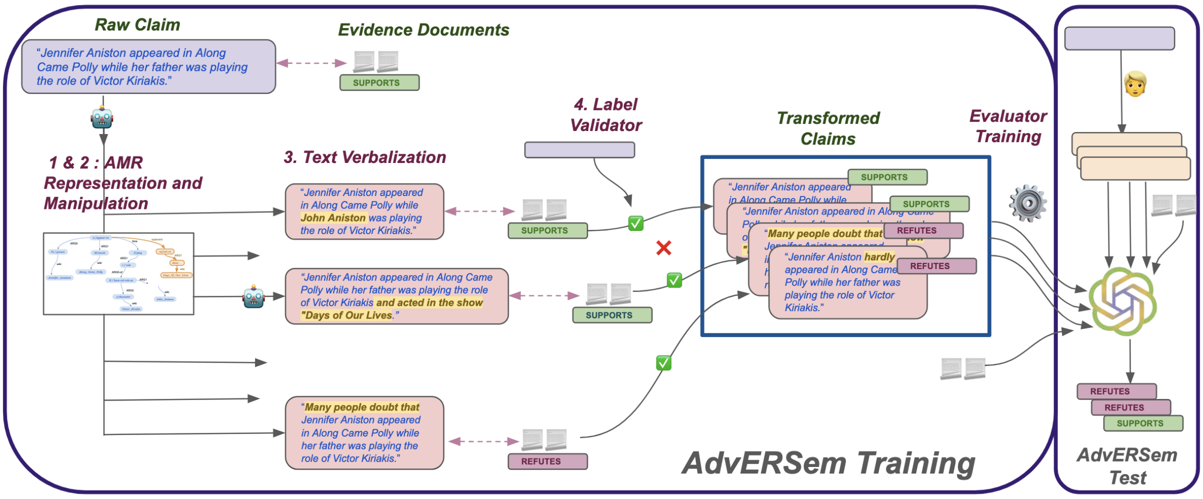

Raw Claim
"Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis."
Evidence Documents
SUPPORTS
1 & 2 : AMR Representation and Manipulation
3. Text Verbalization
"Jennifer Aniston appeared in Along Came Polly while John Aniston was playing the role of Victor Kiriakis."
SUPPORTS
"Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis and acted in the show "Days of Our Lives."
SUPPORTS
"Many people doubt that Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis."
REFUTES
4. Label Validator
Transformed Claims
"Jennifer Aniston appeared in Along Came Polly while
SUPPORTS
"Jennifer Aniston appeared in Along Came
SUPPORTS
"Many people doubt that
REFUTES
"Jennifer Aniston hardly appeared in Along Came Polly while her father was playing the role of Victor Kiriakis."
REFUTES
Evaluator Training
AdvERSem Training
REFUTES
REFUTES
SUPPORTS
AdvERSem Test

Figure 2: **AdvERSEM** training and evaluation on **AdvERSEM-Test**.

versarial claims designed to expose strategic vulnerabilities in LLMs. However, generating effective adversarial attacks remains an open research question, requiring careful strategies to ensure examples both challenging and representative examples.

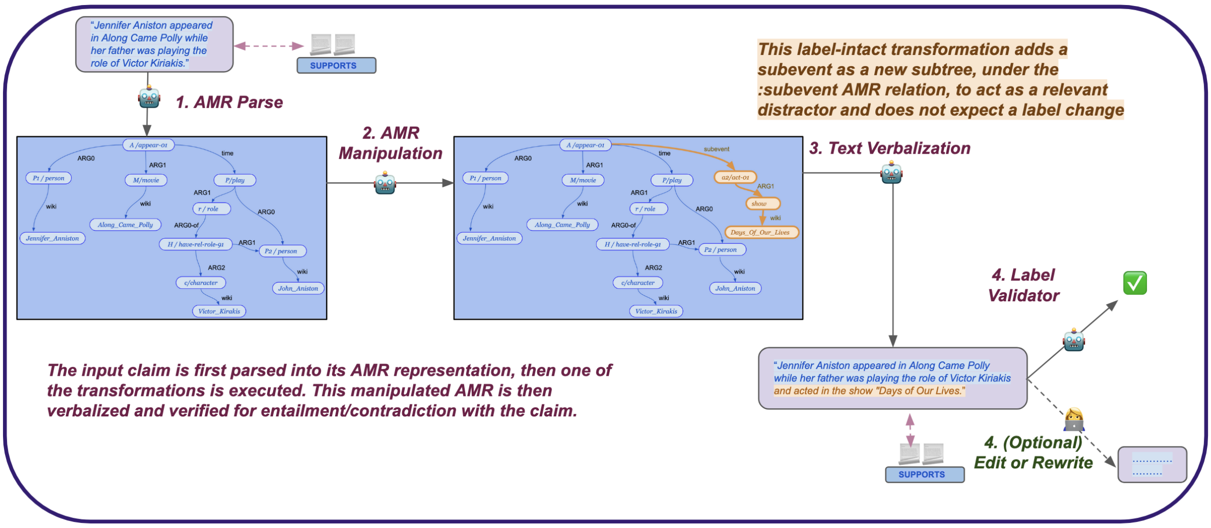
Attempts to create adversarial examples for claim verification have traditionally focused on surface-level perturbations to challenge model robustness (Thorne et al., 2018b). These methods typically involve introducing noise, substituting entities, or making minor lexical alterations to input texts. While such approaches have been instrumental in exploiting LLM vulnerabilities, they often fail to capture actual errors involving deeper semantic and syntactic complexities (Morris et al., 2020) or carefully crafted manipulation like hedging often used to dodge potential disagreement (Hyland, 1998; White, 2003) – for instance journalists may use phrases like "some suggest," "it might appear," or "many doubt" to propose claims while distancing themselves from responsibility or certainty.

Besides, many of these traditional techniques, while revealing some issues related to logical reasoning or factual consistency, rarely provide clarity on specific patterns that evaluators fail to understand. Moreover, these surface-level perturbations can sometimes lead to unnatural or ungrammatical sentences, limiting their effectiveness in real-world applications. To improve groundedness evaluation, it is hence crucial to identify more abstract, structured, and compositional patterns that mimic human language patterns as well as provide fine-grained assessments.

In this paper, we investigate two research questions – ***RQ1***: How can we systematically generate adversarial claims to attack automated groundedness evaluators, and which attacks are particularly effective against SOTA LLM-based evaluators? ***RQ2***: How can we generate useful adversarial data for training groundedness evaluators that are robust to such attacks?

To investigate these research questions, we propose a novel framework that systematically generates adversarial examples through manipulations in a higher-order logical space, particularly through Abstract Meaning Representations (AMR). By extracting away from syntactic variations, AMR provides a structured, graph-based representation of sentence semantics. By manipulating claims in their abstract representations, we can create challenging test cases that expose specific weaknesses in groundedness evaluators, which provide systematic feedback of groundedness performance while addressing the typical issues. In summary, our contributions are the following:

1. We identify a family of semantics-based adversarial claim manipulations that resemble human-like manipulations like hedging, and show that these manipulations can successfully mislead SOTA LLM-based groundedness evaluators.
2. We propose **AdvERSEM** – a framework for **Adv**ersarial **E**valuation and **R**obustness through **Sem**antic **M**anipulations – an effective, and robust framework to evaluate and train LLMs by assessing their groundedness systematically across multiple fine-grained dimensions.
3. We create a manually-curated holistic adversarial test generated from **AdvERSEM** called **AdvERSEM-Test**, with various types of manipulations that mimick real-world groundedness errors.



“Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis.”

This label-intact transformation adds a subevent as a new subtree, under the :subevent AMR relation, to act as a relevant distractor and does not expect a label change

The input claim is first parsed into its AMR representation, then one of the transformations is executed. This manipulated AMR is then verbalized and verified for entailment/contradiction with the claim.

“Jennifer Aniston appeared in Along Came Polly while her father was playing the role of Victor Kiriakis and acted in the show "Days of Our Lives."

Figure 3: Illustration of Adversarial Manipulation using Abstract Meaning Representation (AMR). In this example Specificity of a claim is manimuplated by adding a “:subevent” subgraph.

4. We demonstrate how **AdvERSEM** can be used to identify potential vulnerabilities and be used to improve accuracy and robustness on multiple fact verification benchmarks.

2 Related Work

We now discuss related research to place our contributions in context.

False or adversarial claims designed to mislead human readers, and now machine-reading models, have been long studied and analyzed. Apart from deliberate biases exhibited by journalists and traditional information providers, many frequently employ rhetorical techniques such as hedging to preempt disagreement (Hyland, 1998; White, 2003; Raina et al., 2024) or careful wordplay aimed at misleading or unduly influencing readers. Recently, the problem of misinformation and disinformation has become critically important due to their unprecedented speed and reach, further amplified by digital platforms and recent LLMs. Specifically, Vosoughi et al. (2018) demonstrated that false news spreads significantly faster, farther, and more broadly than truthful information, underscoring the urgent need for improved claim verification methods. Recent studies, such as those by Zhou et al. (2023), have emphasized this growing challenge by illustrating how AI-generated misinformation convincingly integrates fabricated details with truthful elements, effectively evading traditional detection approaches.

Hence, automated evaluation of claims, or fact verification, has been an active area of research. Works such as MultiFC (Augenstein et al., 2019), LIAR (Wang, 2017), and AVeriTeC (Schlichtkrull et al., 2023), emphasize real-world claims verified by journalists or professionals, offering more diverse and context-rich challenges. FEVEROUS (Aly et al., 2021) expands on earlier work by incorporating structured data like tables, while domain-specific datasets like SciFact (Wadden et al., 2022) and COVID-Fact (Saakyan et al., 2021) target scientific or health misinformation.

To improve the robustness of fact verification systems, prior works have produced a range of datasets used to train and evaluate fact-checking systems. Datasets like FEVER (Thorne et al., 2018a), FEVER 2.0 (Thorne et al., 2018b), and VitaminC (Schuster et al., 2021) focus on generating adversarial or subtly false claims to test model robustness, often using Wikipedia as a source, but most of their adversarial generations either use uncontrolled manipulations or use flat first-order logic representations like OpenIE triples (Alonso-Reina et al., 2019). ConQRet (Dhole and Agichtein, 2024; Dhole et al., 2025; Dhole, 2025) provides a benchmark of long-form generations with controlled hallucinations for fine-grained evaluation of groundedness in retrieval-augmented systems.

While previous work has considered adversarial claim generation to evaluate and train fact checking models, our work is the first, to our knowledge, to systematically investigate the specific types of semantic manipulation that can successfully attack state-of-the-art LLM-based fact checkers, and propose a controllable way to generate successful semantic manipulation attacks that can be used to both evaluate existing fact checkers, and to automatically augment training data to make the fact checkers more robust to a range of attacks.

3 Proposed Method: AdvERSEM

We now describe our proposed method, **AdvERSEM** starting with the end-to-end overview of the method and then diving into the details of the implementation for each module.

3.1 Training Fact Verifiers: Overview

Given a model-generated or a human-written claim, $\mathbf{c}$, and evidence text $\mathbf{D}$, the task of groundedness evaluation refers to identifying the veracity $\mathbf{v}$ of the claim, i.e., whether the given evidence supports or refutes the claim, where $\mathbf{v} \in \{\text{SUPPORTS}, \text{REFUTES}\}$.

AdvERSEM (i) parses a claim into its AMR graph, (ii) applies multiple manipulations, (iii) verbalises the edited graphs, and (iv) trains (and evaluates) a claim verifier on the mixed i.e. original + adversarial pairs—see the full pipeline in Figure 2. A sample **AdvERSEM** manipulation (Subevent Addition) is shown in Figure 3.

3.2 Adversarial Manipulation of Claims using Abstract Meaning Representation

We now delve into the details of adversarial generation of claims to augment the testing and training data for evaluator models, namely steps 1 and 2 in Figure 2.

Our approach is to design controllable adversarial attacks based on Abstract Meaning Representations (AMR) (Banarescu et al., 2013). AMR is a semantic representation that represents sentences as rooted, directed, acyclic graphs, abstracting away from syntactic variations. This abstraction allows for systematic manipulations of sentence meaning through graph transformations. We developed 17 specialized AMR-based manipulations aimed at altering claims in various ways. Each manipulation targets a distinct aspect, as detailed in Table 1. The corresponding prompts used to implement these manipulations are provided in Appendix A. For instance,"*Focus Shift*" manipulation changes the entity being focused upon, while "*Hedging*" attempts to introduce mild uncertainty – i.e. either making the claim appear milder or stronger. "*Polarity Negation*" either adds or removes polarity to flip the veracity, while "*SubEvent Addition*" adds a subevent related to the main event.

We manually validate the different verbalisations of each one of the 17 different AMR manipulations, and finalize 10 of those for which veracity is guaranteed – namely *Topic Addition, Source Addition, Subevent Addition, Structural Reversal, Hedging, Focus Shifting, Scalar Adverb Negation, Polarity Negation (either removal or addition), and Scalar Negation*. Details of the remaining ones are provided in Appendix A and will be used in the future for noisy data augmentation.

We now describe **AdvERSEM** in detail. To create an adversarial test set for each of the defined manipulation types, we employed `GPT4-o` (Achiam et al., 2023) to execute each of the following steps:
1. ***AMR Parsing***: We first convert the input claim c into its AMR graph representation G_c using a 10-shot prompt based on the AMR 1.2.6 specification. We employ the few-shot setting owing to its superior performance and conformity to structure in previous settings (Ettinger et al., 2023). This transformation can be denoted as:

$$G_c = \texttt{AMRParse}(c)$$

where $\texttt{AMRParse}(\cdot)$ is the function that maps natural language claims to their semantic graph representations. An example of the parsing prompt is shown in Figure 4.
2. ***AMR Manipulation***: The AMR graph G_c is then modified according to a specified manipulation action a, guided by a natural language instruction, resulting in a new, manipulated AMR $\tilde{G}_c$:

$$\tilde{G}_c = \texttt{Manipulate}(G_c, a)$$

where $\texttt{Manipulate}(\cdot)$ alters the structure while maintaining plausibility and grammaticality. An example of this process is illustrated in Figure 5.
3. ***Text Verbalization***: Finally, the manipulated AMR $\tilde{G}_c$ is verbalized into a new natural language claim $\tilde{c}$ using a 10-shot prompt:

$$\tilde{c} = \texttt{Verbalize}(\tilde{G}_c)$$

where $\texttt{Verbalize}(\cdot)$ is the AMR-to-text generation function.
For both parsing and verbalization, we format the prompts with examples adapted from the official AMR specification.[2] Examples for each transformation and a sample subevent transformation are shown in Table 3 and in Figure 3, respectively.

[2] https://github.com/amrisi/amr-guidelines/blob/master/amr.md

Manipulation	Description	Intact	Flipped
	Manipulations (With Veracity Guarantee)		
Specificity: Topic Addition	Attaches background themes or topics to situate the event with (e.g., topic: “movies”)	✓	
Specificity: Source Addition	Provides spatial or origin-based context for the event (e.g., source: “scientists”)	✓	
Specificity: Subevent Addition	Embeds smaller, related events within the main event to deepen the narrative (e.g., subevent: “local protest”)	✓	
Structure Reversal	Re-attaches entities in new relationships by reversing argument structure (e.g., ARG0-of: “movement leader”)	✓	
Hedging	Introduces uncertainty (mild or strong), e.g., via modal verbs or ‘doubt-01’ (e.g., “might”)	✓	✓
Focus Shift	Changes the focus of the claim (e.g., shifting the subject) (e.g., from “spokesperson” to “organization”)	✓	
Scalar Adverb Negation	Reverses claim polarity by adding or removing scalar adverbs broadly (if positive, weaken; if negative, restore)		✓
Polarity Negation (Addition)	Flips the claim’s veracity by adding a negation marker if absent (e.g., polarity neg add: “did not occur”)		✓
Polarity Negation (Remove)	Reverses negation by removing an existing negation marker (e.g., polarity neg remove: remove “not”)		✓
Specific Scalar Negation	Inserts a scalar negation only when there’s no negation (e.g., “barely noticeable”)		✓
Entity Substitution	Replaces named entities with aliases or alternatives from the same category to preserve or invert the original claim.	✓	✓
Temp/Numeric Attribute Substitution	Modifies time-based or numerical expres- sions to retain or change the truth value of the sen- tence.	✓	✓

Table 1: Adversarial AMR manipulations. A checkmark indicates the veracity setting(s) in which the manipulation applies. The extended list is shown in Appendix Table 7.

3.3 Sentence Based Manipulation

In the same spirit of AMR-based manipulations, we introduced two adversarial manipulations that operate directly over the claim sentences.

Entity Attribute Substitution Replaces named entities with aliases or alternatives from the same category to preserve or flip veracity.

Temporal and Numerical Attribute Substitution Modifies time-based or numerical expressions to retain or flip the veracity.

3.4 Generated Example Validation

To ensure that our generated examples reflect the intended veracity labels, we apply an entailment-based filter that verifies logical consistency between original and transformed claim pairs $(c, \tilde{c})$. Rather than comparing claims against extensive evidence, we use GPT-4o to evaluate entailment directly between concise claim statements.

We provide GPT-4o with two instructions (see Appendix Figure 6): for ***label-intact*** transformations, the original and transformed claims must entail each other; for ***label-flip*** transformations, they must contradict. Only claim pairs that pass this validation are retained. The resulting dataset is then used to train a more robust evaluator model.

The percentage applicability of each transformation is provided in Appendix Table 8.

3.5 Label Agreement

We also measure the agreement between the final annotations and human annotations (Table 2). We find that the agreement rate in the case of label intact is high but poor for the label flipping case, motivating us to create a manually modified evaluation set that we describe in the upcoming section.

	Label Intact	Label Flip
# Agreed Labels	23	13
Total Measured	24	33
Percentage	.96	.40

Table 2: Label agreement for adversarial claims between human annotations and GPT-4o.

4 Experiments

We now present the choice of datasets and models used along with the corresponding experiments.

4.1 Datasets

We evaluate our performance on 3 datasets using 2 metrics. We consider 1) FEVER 2.0 (Thorne et al., 2018b), which provides more realistic adversarial claims, 2) AVERITEC (Schlichtkrull et al., 2023), whose claims necessitate verification using publicly available noisy sources over the web, and 3) introduce **AdvERSEM-Test** our novel manually verified evaluation set. We measure – the accuracy i.e. performance on the raw dataset, and robustness i.e. accuracy on the **AdvERSEM** transformed sets.

The FEVER 2.0 dataset comprises adversarial examples generated by competing systems in the associated shared task and subsequently refined manually by the task organizers. These examples primarily leverage Wikipedia as their source of evidence. For a more realistic evaluation of large language models, particularly Retrieval-Augmented Generation (RAG) systems, we additionally use the AVERITEC fact verification dataset. AVERITEC includes claims supported by links to publicly accessible websites. We augment AVERITEC by scraping these websites using custom Python scripts and incorporate them as evidence.

Finally, we apply the **AdvERSEM** transformations described in §3 to both the training and test sets of FEVER 2.0 and AVERITEC.

We retain only the claims labeled as either *Supported* or *Refuted* in both datasets. From the FEVER dataset, we select 800 examples, reserving 80 for testing and development. For the AVERITEC dataset, we reserve 120 examples for testing and development out of a total of 1,565.

4.2 AdvERSEM-Test

To create **AdvERSEM-Test**, we pass the first 20 examples of the FEVER 2.0 test set, through each of the **AdvERSEM** transformations. The generated claims are then manually refined to ensure that the transformation's particular change is reflected. Publicly available LLM based interfaces are used as an intermediary if needed. Through this process, we gather 200 adversarial and manually verified claims for systematic fine-grained analysis.[3]

4.3 Training Details

We train our groundedness evaluator using GPT-4o-mini (Achiam et al., 2023) and LLaMA-3.1-8B (Dubey et al., 2024), balancing performance and cost. Models are trained for 3 epochs with a batch size of 32 using a supervised chat completion objective. LLaMA-3.1-8B is trained via LLaMAFactory (Zheng et al., 2024) and HuggingFace (Wolf et al., 2020).

Baselines: We used various zero-shot LLMs, including GPT4o, GPT4o-mini, gemini-2.0-flash, and llama-3.1-8B, by leveraging the prompt as shown in Figure 8.

Regular Trained: We trained our groundedness evaluators on the given human-labeled training set (without any adversarial manipulations) *i.e.* $(\mathbf{D}, \mathbf{c}, \mathbf{v})$ tuples.

AdvERSEM Trained: We additionally include **AdvERSEM**-generated adversarial claims in our training set *i.e.,* $(\mathbf{D}, \mathbf{c}, \mathbf{v}) + (\mathbf{D}, \tilde{\mathbf{c}}, \mathbf{v})$ tuples. We also experiment by choosing the number of training examples from the adversarial set in proportion to the errors (EP) on the FEVER 2.0 dev set. Let M be the set of all manipulation types, and E_m the number of errors for type m. Then the probability of selecting an adversarial example of type m is

$$P(m) = \frac{\mathrm{E}_m}{\sum_{k \in M} \mathrm{E}_k}.$$

5 Results

We now present the results for all our evaluations.

5.1 Accuracy and Robustness

The summarised results are present in Table 4 both on the transformed sets (**T**) as well as the raw (**R**) sets of all the fact verification datasets.

We find that **AdvERSEM**-trained models are significantly more robust than regularly trained models, both in terms of macro as well as micro average across all the 3 benchmarks. **For instance, by including AdvERSEM generated examples, the micro-average performance improves by 7.1% in the case of GPT4o-mini, and 3% in the case of Llama-3.1 over the AdvERSEM-Test.**

Additionally, **AdvERSEM** also keeps the performance on other raw datasets like FEVER 2.0 intact. For instance, GPT4o-mini improves the performance on the FEVER 2.0 test set by 1.2% while significantly improving robustness.

5.2 Fine-Grained Robustness Analysis

Table 5 illustrates the fine-grained performance across various adversarial manipulations of **AdvERSEM-Test**. Models trained using the **AdvERSEM** approach consistently outperform both baseline and regularly trained models, demonstrating substantial gains in robustness across nearly all categories of adversarial manipulation. Notably, **GPT4o-mini trained with AdvERSEM achieves the highest macro-averaged robustness of 85%, significantly surpassing its regular training variant by 6.6% and the zero-shot GPT-4o baseline by approximately 9.6%.** Among label-flipped manipulations, the improvement is particularly marked, with the performance on manipulations such as *scalar negation* and *hedging* improving by approximately 25% and 20%, respectively.

When we look at specific manipulations, we observe that **LLMs fail extensively on label-flipped manipulations, specifically on negations, and perform the poorest on hedging**. Besides, random training data is insufficient to mitigate those errors and in fact may also hurt performance. For instance, when evaluated on scalar negatives (*Specific Scalar Negation*), GPT4o-mini reduced performance from 70% to 55% after being trained on FEVER 2.0.

On the other hand, the **AdvERSEM**-trained GPT4o-mini demonstrates pronounced improvements in detecting these adversarial changes, par-

[3] 11 transformations were applied on all the 20 examples, in which 2 transformations, viz., *Polarity Negation Removal* and *Polarity Negation Addition* were applicable on 17 and 3 examples respectively, as only 3 out of 20 claims possessed a negation in their raw form.

Transformation	**Claim (original → transformed)**
Specificity: Topic Addition	In 2010, the population of Europe was larger than 61 → more than 61 million, according to demographic trends.
Specificity: Source Addition	The Woman in Black was abandoned by Hammer Film Productions in the 2010s in favor of working on Freddie vs. Jason , according to industry insiders.
Specificity: Subevent Addition	"Honeymoon" is the second major-label record by Elizabeth Woolridge Grant, and it was released in 2015.
ARG0-of	There exists a Korean band called Scandal is a band from Korea.
Hedging	People don't doubt that "Excuse My French" is the debut album of Karim Kharbouch(French Montana).
Focus Shift	There is not a natural element that goes by the name of Moscovium does not exist naturally.
Scalar Adverb Negation	Eurotas is definitely not a minor river of Laconia.
Polarity Neg Addition	The lead engineer of the iAPX 432 did not work at Intel for 20 more years → more than 20 years after its introduction.
Polarity Neg Removal	Dawood Ibrahim Kaskar was → is not from a place in Mira-Bhayandar, Thane district.
Scalar Negation	Exotic Birds hardly rejected to be → being an opening band for a band from Minneapolis.
Hedging	The general public doubts that Andrew Kevin Walker was born on August 14, 1864 and is a screenwriter.

Table 3: Transformation examples for each type on FEVER 2.0 claims – pieces of text removed from the original claim are shown in red while those added in the new claim are shown in green. The upper half shows Veracity Intact ones, while the bottom half shows Veracity Flipped ones.

	Manually Verified Evaluation Sets			Automatically Created Evaluation Sets				
	AdvERSEM-Test (T)		**FEVER 2.0 (R)**	**AVERITEC (R)**	**AVERITEC (T)**		**FEVER 2.0 (T)**	
Model	Macro	Micro	R	R	Macro	Micro	Macro	Micro
Llama-3.1-8B	**.792**	**.766**	.652	.680	**.588**	.653	.498	.504
gpt-4o-mini	.763	.762	.821	.786	.569	.642	.517	.550
gpt4o	.754	.751	**.833**	**.880**	.585	**.675**	**.556**	**.579**
gemini-2.0-flash	.761	.755	.731	.788	.562	.613	.515	.532
gpt-4o-mini (Regular Trained)	.784	.793	.885	-	-	-	.569	.601
gpt-4o-mini (**AdvERSEM** Trained)	**.850**	**.864**	**.897**	-	-	-	**.585**	**.625**
llama-3.1-8B (Regular Trained)	.745	.749	**.805**	**.817**	.599	.652	.516	.526
llama-3.1-8B (**AdvERSEM** Trained)	**.771**	**.779**	**.805**	.800	**.631**	**.676**	**.518**	**.532**

Table 4: Performance on FEVER 2.0, AVERITEC, and **AdvERSEM-Test**. **R**=raw/original set, **T=AdvERSEM** transformed sets. Best scores across models are highlighted in bold. Top set of rows represent zero-shot variants.

Approach \ Label Change	Label Intact						Label Flipped					Overall	
	Topic	***Source***	***Subevent***	***ARG0-of***	***Hedging***	***Focus Shift***	***Scalar Adverb Neg***	***Polarity Neg Add***	***Polarity Neg Rem***	***Specific Scalar Negation***	***Hedging***	***Macro Average***	***Micro Average***
#Examples	20	20	20	20	20	20	20	17	3	20	20	200	200
llama-3.1-8B	.850	.737	.850	.750	.800	.789	.526	.588	.333	.500	.450	.652	.680
gpt-4o-mini	.900	.800	.900	.850	.900	.750	.684	.647	.667	.700	.600	.763	.773
gpt-4o	.900	.800	.900	.950	.900	.900	.526	.647	.667	.600	.500	.754	.763
gemini-2.0-flash	.850	.800	.850	.800	.850	.800	.667	.882	.667	.600	.600	.761	.767
llama-3.1-8B (Regular Trained)	.750	.700	.800	**.900**	**.800**	.650	.737	**.941**	.667	**.650**	.600	.745	.749
llama-3.1-8B (**AdvERSEM** Trained)	**.800**	**.750**	**.850**	.850	.750	**.850**	**.789**	.824	.667	.600	**.750**	**.771**	**.779**
gpt-4o-mini (Regular Trained)	**.950**	.850	.900	**.950**	**.900**	.750	.684	.824	.667	.550	.600	.784	.793
gpt-4o-mini (**AdvERSEM** Trained)	**.950**	**.900**	**.950**	.850	**.900**	**.800**	**.789**	**.941**	.667	**.800**	**.800**	**.850**	**.864**

Table 5: Fine-grained performance on **AdvERSEM-Test** demonstrating increased robustness.

ticularly in categories like *scalar adverb negation* (+10.5%), *polarity negation addition* (+11.7%), and *scalar negation* (+25%). Similar enhancements are observed with llama-3.1-8B, reinforcing that training with semantically structured adversarial examples notably boosts the robustness of models across various semantic alterations. These results indicate that structured AMR-based adversarial training generalizes well across different linguistic adversarial manipulations as well as significantly mitigates vulnerabilities to challenging alterations commonly employed in deceptive claims.

5.3 Analysis on FEVER and AVERITEC

We now look at the detailed performances over FEVER 2.0 and AVERITEC (shown in Appendices Table 10 and Table 11). We observe that the average robustness for all the zero-shot models is low. While utilizing FEVER 2.0 training data improves performance on the raw test set, it only gives a slight boost in robustness. When **AdvERSEM**-based examples are used for training, the robustness improves further, maintaining accuracy.

Across both FEVER 2.0 and AVERITEC, we observe a substantial drop in accuracy for all models on adversarially manipulated claims, particularly those that challenge deeper aspects of semantics (e.g., focus shift, structural reversal, or hedging). For example, **zero-shot GPT4o and gemini-2.0-flash experience significant drops in performance on manipulations like scalar adverb negation and polarity reversals**.

Zero-shot models, despite strong results on standard test sets, saw significant performance degradation on certain adversarial manipulations, especially those involving negations – e.g., GPT-4o's accuracy drops to 0.286 on scalar adverb negations.

On AVERITEC, which features noisy, real-world web claims, adversarially trained models retain accuracy even in the presence of more diverse and noisy evidence, showing the practical benefit of these techniques beyond synthetic testbed.

Proportion of Training Examples Used	Approach	Accuracy
0	gpt4o	.833
	gemini-2.0-flash	.731
	gpt-4o-mini	.821
1/3	gpt-4o-mini (FEVER 2 trained)	.800
	gpt-4o-mini (AdvERSEM Trained)	**.829**
1/2	gpt-4o-mini (FEVER 2 trained)	.859
	gpt-4o-mini (AdvERSEM Trained)	**.883**
1	gpt4o-mini (FEVER 2 trained)	.897
	gpt4o-mini (**AdvERSEM** Trained)	.863
	gpt4o-mini (AdvERSEM Trained EP)	**.897**

Table 6: Accuracy of different models trained on smaller subsets of FEVER 2.0 and evaluated on FEVER 2.0 Test split. Note that AMR-generated adversarial examples can improve raw performance with lesser number of annotated examples. (EP = Error Proportions)

5.4 Out-of-Domain Robustness

We also evaluate the out-of-domain robustness of these models by training on FEVER 2.0 and its **AdvERSEM** manipulations, then testing on the AVERITEC dataset.

Our results in Appendix Table 12 show that while zero-shot and regular finetuned models (such as GPT4o-mini and its FEVER 2-trained variant) maintain competitive accuracy on label-intact AMR manipulations, their robustness substantially declines on label-flipped and sentence-based adversarial attacks—especially for semantically challenging manipulations such as scalar negation or polarity reversal. Notably, adversarially finetuned models (i.e., those further trained on AMR-generated adversarial data) consistently outperform their non-adversarially trained counterparts across both AMR-based and sentence-level manipulations. For example, GPT4o-mini with AMR-based adversarial training achieves an average robustness of 0.674, outperforming the FEVER 2-trained (0.621) and vanilla (0.611) baselines, improving AMR robustness from 0.652 to 0.712. These results highlight that adversarial AMR-based training boosts LLM resilience to a broad spectrum of semantic manipulations, as well as generalizes well in out-of-domain settings.

5.5 AdvERSEM Training Sample Efficiency

We further demonstrate the effectiveness of incorporating **AdvERSEM**-based adversarial training in low-resource settings. We adjusted the amount of FEVER 2.0 training data available to the model and controlled the portion used for generating AMR adversarial examples—ranging from zero-shot (0) to one-third (1/3), half (1/2), and the full (1) dataset—and assessed performance on the FEVER 2.0 test set. Consistently, models enhanced with AMR-generated adversarial examples matched or exceeded the performance of those trained exclusively on human-annotated claims across all settings. The results are described in detail in Table 6.

The benefit of our approach is especially pronounced when labeled data is scarce. For example, GPT4o-mini trained with AMR-generated adversarial examples consistently outperforms its zero-shot counterpart, without signs of overfitting. In contrast, training with human-labeled FEVER 2.0 data alone leads to a 3% drop in performance when only one-third of the data is used. Moreover, expanding the proportion of AMR-based training data to one-third and half substantially closes the performance gap with the full-data baseline (0.897) by 11% and 82%, respectively. These findings underscore the sample efficiency of AMR-based augmentation, which introduces diverse and informative adversarial variations that enhance generalization, even with limited annotations.

6 Conclusions and Future Work

We introduced **AdvERSEM**, a novel framework that leverages Abstract Meaning Representations (AMR) to systematically evaluate and improve the robustness of groundedness evaluators through structured adversarial data augmentation. We uncover several vulnerabilities by introducing semantically controlled, fine-grained manipulations of claims. These adversarial manipulations expose specific weaknesses in existing evaluators and provide interpretable and actionable feedback beyond single-dimensional test scores, highlighting the importance of robustness and interpretability in groundedness evaluation, and also serve as a caution on agent-style modular systems, which predominantly rely on LLMs in a zero-shot manner.

Moreover, training groundedness evaluators using our structured adversarial examples significantly enhances their ability to withstand complex semantic perturbations such as hedging, negation, and specificity adjustments, addressing key vulnerabilities identified in LLMs as well as human writings. By generating challenging yet realistic training examples, our approach effectively reduces reliance on expensive annotated data, thereby facilitating efficient and robust model development. The **AdvERSEM** framework, and associated evaluation set **AdvERSEM-Test** can be readily extended to various other NLP applications, for systematically testing and enhancing model reliability.

7 Limitations

While **AdvERSEM** provides a structured and interpretable framework for systematically assessing and improving the robustness of groundedness evaluators, there are several limitations worth noting.

Our adversarial examples, though systematically designed and verified, are generated through a large language model (GPT-4o), which itself might introduce unintended biases or noise (Mitchell et al., 2025). Although we employed entailment checks and manual validation steps to mitigate these risks, some residual inaccuracies could persist. And hence, to provide a reliable estimate of the behavior of these groundedness evaluators, we manually modified split on top of the same.

AdvERSEM encompasses AMR parsing and AMR verbalization for English text. While AMR parsing has been expanded to many languages (Soto Martinez et al., 2024; Kang et al., 2024), cross-lingual and multi-lingual parsing is still an active area of research (Mansouri, 2025), and our method would need to be evaluated for those languages separately.

Additionally, while our method significantly improves robustness against specific adversarial manipulations, it might not cover all possible adversarial strategies, particularly those exploiting multi-sentence coherence or higher-level rhetorical manipulations. Future work could expand the diversity of manipulations and further explore the integration of human-generated adversarial examples to address these gaps comprehensively.

8 Ethics Statement

Our work operates within the broader context of combating misinformation, as large language models (LLMs) can be exploited for malicious purposes. Therefore, developing accurate, reliable, and robust assessment methods is essential. Systematically enhancing claim verification and groundedness evaluation is crucial for countering increasingly sophisticated misinformation and disinformation tactics, especially as LLM-generated simulations become proliferate (Dhole, 2024, 2023).

However, while recent advancements in LLM-based evaluations have shown promise, our research highlights that these models remain brittle and susceptible to exploitation. Consequently, research like ours, which is situated within fact verification and related domains, must always be supported by rigorous manual evaluation, particularly to ensure robustness. **AdvERSEM-Test**, is specifically created with this consideration in mind.

We used GPT4o and Grammarly to help improve the grammar of the text and for creating LaTeX outlines for tables and images.

Acknowledgments

This work was partially supported by the Microsoft Accelerate Foundation Models Award. The authors also thank the anonymous reviewers for their insightful comments.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Aimée Alonso-Reina, Robiert Sepúlveda-Torres, Estela Saquete, and Manuel Palomar. 2019. Team gplsi. approach for automated fact checking. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 110–114.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and verification over unstructured and structured information (feverous) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. Multifc: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Kaustubh Dhole. 2023. Large language models as sociotechnical systems. In *Proceedings of the Big Picture Workshop*, pages 66–79.

Kaustubh Dhole. 2024. Kaucus-knowledgeable user simulators for training large language models. In *Proceedings of the 1st Workshop on Simulating Conversational Intelligence in Chat (SCI-CHAT 2024)*, pages 53–65.

Kaustubh Dhole and Eugene Agichtein. 2024. Llm judges for retrieval augmented argumentation.

Kaustubh Dhole, Kai Shu, and Eugene Agichtein. 2025. Conqret: A new benchmark for fine-grained automatic evaluation of retrieval augmented computational argumentation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5687–5713.

Kaustubh D Dhole. 2025. To retrieve or not to retrieve? uncertainty detection for dynamic retrieval augmented generation. *arXiv preprint arXiv:2501.09292*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models.

Allyson Ettinger, Jena Hwang, Valentina Pyatkin, Chandra Bhagavatula, and Yejin Choi. 2023. "you are an expert linguistic annotator": Limits of LLMs as analyzers of Abstract Meaning Representation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8250–8263, Singapore. Association for Computational Linguistics.

Ken Hyland. 1998. Hedging in scientific research articles.

Jeongwoo Kang, Maximin Coavoux, Cédric Lopez, and Didier Schwab. 2024. Should cross-lingual AMR parsing go meta? an empirical assessment of meta-learning and joint learning AMR parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 43–51, Miami, Florida, USA. Association for Computational Linguistics.

Dongryeol Lee, Yerin Hwang, Yongil Kim, Joonsuk Park, and Kyomin Jung. 2025. Are LLM-judges robust to expressions of uncertainty? investigating the effect of epistemic markers on LLM-based evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8962–8984, Albuquerque, New Mexico. Association for Computational Linguistics.

Behrooz Mansouri. 2025. Survey of abstract meaning representation: Then, now, future. *arXiv preprint arXiv:2505.03229*.

Margaret Mitchell, Giuseppe Attanasio, Ioana Baldini, Miruna Clinciu, Jordan Clive, Pieter Delobelle, Manan Dey, Sil Hamilton, Timm Dill, Jad Doughman, et al. 2025. Shades: Towards a multilingual assessment of stereotypes in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11995–12041.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.

Ananjan Nandi, Christopher D Manning, and Shikhar Murty. 2025. Sneaking syntax into transformer language models with tree regularization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8006–8024, Albuquerque, New Mexico. Association for Computational Linguistics.

Amie Paige, Adil Soubki, John Murzaku, Owen Rambow, and Susan E. Brennan. 2024. Training LLMs to recognize hedges in dialogues about roadrunner cartoons. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 204–215, Kyoto, Japan. Association for Computational Linguistics.

Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is LLM-as-a-judge robust? investigating universal adversarial attacks on zero-shot LLM assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7499–7517, Miami, Florida, USA. Association for Computational Linguistics.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. Covid-fact: Fact extraction and verification of real-world claims on covid-19 pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2116–2129.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web. *Advances in Neural Information Processing Systems*, 36:65128–65167.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

William Soto Martinez, Yannick Parmentier, and Claire Gardent. 2024. Generating from AMRs into high and low-resource languages using phylogenetic knowledge and hierarchical QLoRA training (HQL). In *Proceedings of the 17th International Natural Language Generation Conference*, pages 70–81, Tokyo, Japan. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *science*, 359(6380):1146–1151.

David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. 2022. Scifact-open: Towards open-domain scientific claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4719–4734.

William Yang Wang. 2017. “liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426.

Peter RR White. 2003. Beyond modality and hedging: A dialogic view of the language of intersubjective stance. *Text & Talk*, 23(2):259–284.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI ’23, New York, NY, USA. Association for Computing Machinery.

A AMR Manipulations

In Table 7, we show our extended set of AMR manipulations. This list shows the remaining manipulations that may not guarantee veracity. For instance, in "*Quantifier Alteration*", when we change the scope or amount expressed in the claim, the new claim may not always maintain (or always flip) the veracity.

B Robustness Analysis on the Dev Set

In Table 9, we present the robustness analysis on the development set of FEVER 2.0 using GPT4o-mini. The fall in accuracy is calculated as the relative decline in accuracy when evaluated with the transformed dev set as compared to the raw dev set. We then use the inverse of the fall of accuracy, which is used to dictate the proportion of adversarial examples that would be used in training.

```
You are an advanced semantic parser. Read a sentence and
produce only its AMR or Abstract Meaning Representation,
without explanations.
Below is an example from the AMR 1.2.6 Specification.

EXAMPLE:
Sentence: Patrick Makau finished the marathon in 2 hours.
AMR:
(f / finish-01
    :ARG0 (p / person :wiki "Patrick_Makau_Musyoki"
           :name (n / name :op1 "Patrick" :op2 "Makau"))
    :ARG1 (r / run-02
        :ARG0 p
        :ARG1 (m / marathon)
        :duration (s2 / sum-of
            :op1 (t2 / temporal-quantity
                :quant 2
                :unit (h / hour)))))
...
...
Now convert the input sentence to AMR. Return only the AMR.
Sentence: {text}
AMR:
```

Figure 4: Text-to-AMR prompt example used for parsing original claims.

Figure 7 shows the robustness trends with and without adversarial examples.

C Sample Efficiency: Training with limited data

The Figure 7 illustrates the sample efficiency of our approach, highlighting its robustness across varying numbers of original examples.

D Prompt Templates

We provide the prompt templates used in each step of our framework. Figure 4, 5, and 6 correspond to the key steps: AMR parsing, AMR manipulation, and generated example validation, respectively.

```
The following is an AMR or Abstract Meaning Representation
of a claim {claim} which is {label} by evidence:

Please perform the following action to create a new AMR
whose verbalisation would be different:

The action should make the new AMR represent a new natural
looking, grammatical, and sensible claim, and also
different from the
original claim. If there is no previous AMR,
then only return
'NO CHANGE'.

Return the new AMR.
Previous AMR: {AMR}
Label: {The new Label}
Description: {Description of Manipulation}
Action: {Action Name of the Manipulation}

New AMR:
```

Figure 5: Prompt template for AMR manipulation based on graph operations and label changes.

```
"The following two statements are intended to
not contradict each other. "
"Check if they are logically consistent and
do not contradict each other.\n\n"
"Original statement: {}
\n\nNew statement: {}\n\n"
"Respond with 'True' if they do not contradict each other, "
"or 'False' if they do contradict.
Do not respond with anything else."

--------------------------------------------------------------

"The following two statements are expected to reflect
contradictory positions. "
"Check if they indeed contradict each other.\n\n"
"Original statement: {}
\n\nRevised statement: {}\n\n"
"Respond with 'True' if they contradict each other, "
"or 'False' if they do not contradict. Do not respond with
anything else."
```

Figure 6: Entailment and Contradiction

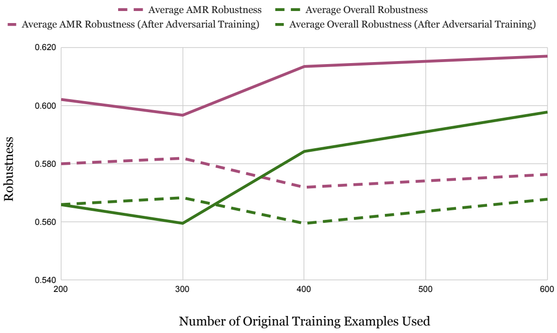


Figure 7: Robustness Trends of With and Without Including Adversarial Examples. At each point, the adversarial examples are generated for the same number of original examples (x-axis)

Manipulation	Description
Modality Shifting	Weaken or strengthen the speaker's commitment to the claim. (e.g., Replace "must" with "might".)
Quantifier Alteration	Change the scope or amount expressed in the claim. (e.g., Alter "100" to "10".)
Presupposition Removal	Delete or contradict a background assumption (e.g., Remove an assumed ongoing subevent.)
Implicature Disruption	Break implied event ordering or assumptions of contrast. (e.g., Replace causal relation with flat conjunction "and")
Rhetorical Question Framing	Transform into a rhetorical question that implies doubt. (e.g., Change to "Did it really..")
Parataxis vs Hypotaxis	Change logical or causal relationships to a flat structure, altering implications. (e.g., Replace "because" with "and")
Figurative Interpretation	Replace with metaphorical or sarcasm (e.g., Replace "bulldoze" with "dominate" to imply sarcasm.)

Table 7: Adversarial AMR manipulations. These manipulations do not guarantee the veracity of the transformed claim.

	Dataset	**AMR Transformations (AdvERsem-FEVER) (Automatic)**				**AMR Transformations (AdvERsem-AVERITEC) (Automatic)**				**AdvERSem-Test (Manual)**
	Split	**Train**		**Dev**		**Train**		**Dev**		**Test**
		TA (%)	**TAv (%)**	**TA (%)**	**TAv (%)**	**TA (%)**	**TAv (%)**	**TA (%)**	**TAv (%)**	**TA (%)**
Label Intact	Specificity: Topic Addition	99.84	68.90	100	79.75	100.00	80.96	100.00	84.40	100
	Specificity: Source Addition	100.00	71.61	100	81.01	99.85	78.15	100.00	81.65	100
	Specificity: Subevent Addition	99.84	64.75	100	69.62	99.62	77.62	100.00	72.48	100
	Structure Reversal	100.00	67.46	98.73	70.89	99.85	71.47	100.00	72.48	100
	Hedging	99.84	75.92	100	77.22	100.68	77.62	99.08	76.15	100
	Focus Shift	99.84	67.94	100	69.62	100.08	74.81	100.00	71.56	100
Label Flipped	Scalar Adverb Negation	99.84	25.20	100	36.71	100.23	33.54	101.83	25.69	100
	Polarity (Negative Addition)	68.42	7.18	70	7.59	85.43	6.68	63.30	2.75	85
	Polarity (Negative Removal)	30.30	3.35	28	3.80	12.75	2.73	14.68	1.83	15
	Adding Other Scalar Negations	99.52	37.16	100	39.24	100.08	27.62	101.83	33.94	100
	Hedging	100.00	63.48	100	67.09	100.76	62.82	100.92	58.72	100

Table 8: Transformation Applicability of each transformation without (TA) and with verifier (TAv).

Manipulation Type	**Veracity Change**	**Manipulation Category**	**Fall in Accuracy**
	Raw		- (.810)
AMR Based	**Veracity Intact**	Specificity: Topic Addition	7.91 (.746)
		Specificity: Source Addition	20.92 (.641)
		Specificity: Subevent Addition	7.98 (.745)
		Structure Reversal	11.83 (.714)
		Hedging	12.99 (.705)
		Focus Shift	5.74 (.764)
	Veracity Flipped	Scalar Adverb Negation	31.90 (.552)
		Polarity Negation (Addition)	17.71 (.667)
		Polarity Negation (Remove)	58.85 (.333)
		Scalar Negation	24.34 (.613)
		Hedging	27.80 (.585)
Sentence Based	**Veracity Intact**	Entity Attribute Substitution	25.06 (.607)
		Temporal Numerical Substitution	29.46 (.571)
	Veracity Flipped	Entity Attribute Substitution	31.72 (.553)
		Temporal Numerical Substitution	64.73 (.286)

Table 9: "Fall in Accuracy" indicates the difference from 0.810, with the actual accuracy shown in parentheses. *Fewer examples were available for these negation-based transformations.

```
Classify whether the given evidence SUPPORTS or REFUTES
the given claim.

Evidence: {wiki_text}

Claim: {claim}

Answer (SUPPORTS or REFUTES):
```

Figure 8: Fact Verification instruction used to evaluate the veracity of the claims

		AMR Based Transformations												Sentence Transformations					
Approach \ Label Change		Label Intact						Label Flipped						Label Intact		Label Flipped			
	FEVER	**Topic**	**Source**	**Subevent**	**ARG0-of**	**Hedging**	**Focus Shift**	**Scalar Adverb Neg**	**Polarity Neg Add**	**Polarity Neg Rem**	**Specific Scalar Negation**	**Hedging**	**AMR Average**	**Entity Attribute**	**Temp-Numer**	**Entity Attribute**	**Temp-Numer**	**Sentence Average**	**Overall Average**
#Examples	78	60	59	57	52	62	59	21	21	50	31	49	-	54	58	44	42	-	-
gpt4o	.833	.717	.661	.702	.692	.613	.831	.286	**.500**	.000	.484	.510	.545	.630	**.500**	.500	**.714**	**.586**	.556
gemini-2.0-flash	.731	.672	.586	.625	.635	.610	.746	.429	.333	.000	.516	**.633**	.526	.612	.300	.452	.571	.485	.515
gpt-4o-mini	.821	.717	.678	.719	.731	.629	.831	.429	.000	.000	**.581**	.449	.524	.556	.300	.568	.571	.499	.517
llama-3.1-8B (FEVER 2 Trained)	**.702**	.674	**.609**	**.725**	**.634**	.568	.667	.500	.200	.000	.381	**.643**	.509	.486	.250	**.588**	.667	.498	.506
llama-3.1-8B (**AdvERSEM** Trained)	.698	**.702**	.587	.718	**.634**	**.581**	**.689**	**.643**	**.200**	.000	**.429**	**.677**	**.533**	.486	.250	.500	.667	.476	**.518**
gpt-4o-mini (FEVER 2 trained)	.885	.800	.695	.719	.846	.710	**.881**	.381	.333	.000	.484	.490	.576	.660	.400	.545	.571	.544	.568
gpt-4o-mini (**AdvERSEM** Trained)	**.897**	**.817**	**.746**	**.842**	**.827**	**.726**	.847	**.429**	.167	.000	**.548**	**.551**	**.591**	**.796**	**.500**	.409	.571	**.569**	**.585**

Table 10: Accuracy of different models on transformed adversarial test splits of FEVER 2.0 (Thorne et al., 2018a). Best scores across models are highlighted in bold.

		AMR Based Transformations												Sentence Transformations					
Approach \ Label Change		Label Intact						Label Flipped						Label Intact		Label Flipped			
	AVERITEC	**Topic**	**Source**	**Subevent**	**ARG0-of**	**Hedging**	**Focus Shift**	**Scalar Adverb Neg**	**Polarity Neg Add**	**Polarity Neg Rem**	**Specific Scalar Negation**	**Hedging**	**Average AMR**	**Entity Attribute**	**Temp-Numer**	**Entity Attribute**	**Temp-Numer**	**Sentence Average**	**Overall Average**
#Examples	120	91	89	83	68	89	76	24	33	81	36	63	-	90	70	47	56	-	-
llama-3.1-8B	.766	.765	.813	.773	.789	.684	.738	.250	.375	.667	.226	**.327**	.583	.859	.793	.386	.375	.603	**.588**
gpt-4o-mini	.786	.807	.818	.775	.773	.816	.716	**.333**	.000	.667	**.457**	.290	**.587**	.841	.733	.311	.200	.521	.569
gpt4o	**.880**	**.839**	**.864**	**.863**	**.785**	**.826**	**.811**	.043	.125	.667	.235	.290	.577	**.852**	**.833**	**.444**	.300	**.608**	.585
gemini-2.0-flash	.788	.761	.741	.800	.781	.753	.736	.235	**.500**	.333	.278	.279	.563	.764	.655	.438	**.375**	.558	.562
llama-3.1-8B (Averitec Trained)	**.817**	**.769**	.775	.771	**.721**	.719	.724	.458	.250	.667	.361	.238	.587	.767	**.839**	.319	**.300**	.556	.579
llama-3.1-8B (**AdvERSEM** Trained)	.800	.747	**.787**	**.819**	.676	**.798**	**.776**	**.625**	**.375**	.667	**.639**	**.683**	**.690**	**.789**	.806	**.447**	.200	**.561**	**.656**

Table 11: Accuracy of different models on transformed adversarial test splits of AVERITEC dataset (Schlichtkrull et al., 2023). Best scores across models are highlighted in bold.

	AMR Based Transformations											Sentence Transformations					
Approach \ Label Change	Label Intact						Label Flipped					Label Intact		Label Flipped			
	Topic	**Source**	**Subevent**	**ARG0-of**	**Hedging**	**Focus Shift**	**Scalar Adverb Neg**	**Polarity Neg**	**Specific Scalar Negation**	**Hedging**	**AMR Robustness**	**Entity Attribute**	**Temp-Numer**	**Entity Attribute**	**Temp-Numer**	**Sentence Robustness**	**Average Accuracy**
gpt-4o-mini	.814	.823	.761	.752	.798	**.770**	.407	.468	.386	.319	.630	**.836**	.734	.316	.356	.561	.611
gpt-4o-mini (FEVER 2 trained)	**.864**	**.886**	**.850**	**.818**	**.816**	.743	.292	.375	.486	.387	.652	.795	**.800**	**.356**	.200	.538	.621
gpt-4o-mini (AdvERSEM Trained)	.818	.830	.813	.788	.724	.757	**.583**	**.500**	**.629**	**.677**	**.712**	.818	.767	.289	**.400**	**.568**	**.674**

Table 12: Accuracy of **AdvERSEM** trained models on Out-of-domain data, with FEVER 2.0 adversarial examples and evaluated on AVERITEC test split (Schlichtkrull et al., 2023), broken down by AMR manipulation type, highlighting robustness to out-of-domain adversarial variations.