Sentence Smith: Controllable Edits for Evaluating Text Embeddings

Hongji Li **Andrianos Michail** **Reto Gubelmann** **Simon Clematide** **Juri Opitz**
University of Zurich
Correspondence: opitz.sci@gmail.com

Abstract

Controllable and transparent text generation has been a long-standing goal in NLP. Almost as long-standing is a general idea for addressing this challenge: Parsing text to a symbolic representation, and generating from it. However, earlier approaches were hindered by parsing and generation insufficiencies. Using modern parsers and a safety supervision mechanism, we show how close current methods come to this goal. Concretely, we propose the SENTENCESMITH framework for English, which has three steps: 1. Parsing a sentence into a semantic graph. 2. Applying human-designed semantic manipulation rules. 3. Generating text from the manipulated graph. A final entailment check (4.) verifies the validity of the applied transformation. To demonstrate our framework's utility, we use it to induce hard negative text pairs that challenge text embedding models. Since the controllable generation makes it possible to clearly isolate different types of semantic shifts, we can evaluate text embedding models in a fine-grained way, also addressing an issue in current benchmarking where linguistic phenomena remain opaque. Human validation confirms that our transparent generation process produces texts of good quality. Notably, our way of generation is very resource-efficient, since it relies only on smaller neural networks.

1 Introduction

How can we transform the meaning of a sentence such that the output remains fluent while the transformation process is maximally transparent? Some symbolic methods, e.g., those based on word replacement using taxonomy lookups (Bolshakov and Gelbukh, 2004; Huang et al., 2009), provide transparency and control but often produce unnatural sentences with limited variation. On the other end, of course, we now have the all-dominating paradigm of LLM prompting. But clearly this process is opaque, and any control is, at best, indirect (Greenblatt et al., 2024; Mizrahi et al., 2024).

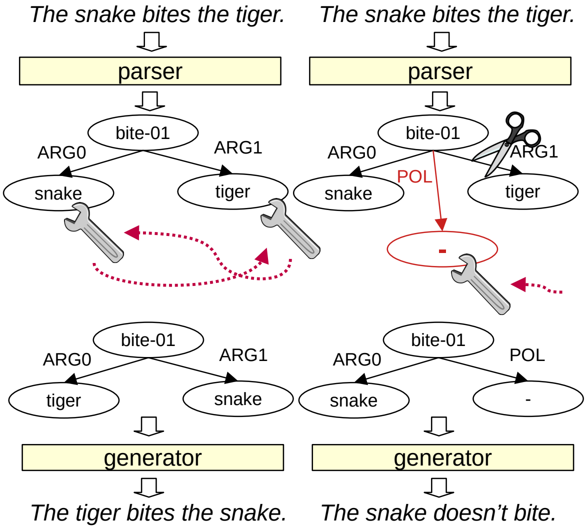

Figure 1: Example application of two controlled text transformations. **Left**: Switching semantic roles in a sentence, where the agent (ARG0: the biter) becomes the patient (ARG1: the bitten), and vice versa. **Right**: Deleting the patient and negating the main predicate.

To bridge the gap between these paradigms, and to highlight an alternative to LLM-based methods, we propose SENTENCESMITH, a neuro-symbolic framework for text manipulation. The process begins with a parser that maps a sentence onto a symbolic graph-based meaning representation, such as an Abstract Meaning Representation (AMR, Banarescu et al., 2013). We posit that such a representation provides an effective interface for applying well-defined, precise, and targeted meaning-altering operations. Once the graph is updated, a generator converts it back into natural text.

The process is illustrated in Figure 1, where a single sentence is transformed into two new sentences with different meanings. On the left, we reverse semantic roles (e.g., *tiger*, *snake*), modifying the event structure of the sentence by swapping patient and agent roles. On the right, we apply a two-step process: removing the patient (*tiger*) and

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 26428–26445
November 4-9, 2025 ©2025 Association for Computational Linguistics

negating the main predicate (*bite-01*), resulting in a new sentence that describes a "harmless" snake. Not shown in Figure 1 but discussed later, we also introduce an optional faithfulness check as a post-processing step to evaluate the consistency between the transformed graph and the generated sentence, filtering out outputs affected by any eventual parsing or generation errors.

We argue that this combination of symbolic control, neural fluency, and post-hoc verification provides a powerful and flexible approach to sentence meaning transformation, offering a more controllable alternative or complement to LLMs.

To illustrate the practical value of SENTENCESMITH, we conduct a detailed demonstration study, leveraging our framework to generate hard negative pairs that pose significant challenges for text embedding models. By applying well-defined transformation rules targeting specific semantic phenomena, SENTENCESMITH creates nuanced sentence pairs that go beyond superficial lexical differences, enabling a more fine-grained evaluation of embedding models. Through this application study, we shed light on their strengths and weaknesses in handling particular linguistic phenomena, addressing a limitation in current benchmarking practices, where such intricacies tend to be obscured.

Broadly speaking, this study contributes to the ongoing effort to pinpoint the limits of Transformer-based Neural Networks' understanding of sentence meaning. It has been established that even the latest LLMs struggle with important sentence-level meaning phenomena such as negation (Parmar et al., 2024), and Xu et al. (2025) finds that LLMs still struggle with exact, syntax-based modes of reasoning, sometimes being outperformed by BART (Lewis et al., 2020), a much smaller and simpler sequence-to-sequence transformer. The fine-grained and dynamic experimental settings enabled by SENTENCESMITH can shed new light on this central, but still open question regarding the abilities of Transformers.

The remainder of this paper is structured as follows: Section §2 reviews background and related work. Section §3 introduces our neuro-symbolic SENTENCESMITH framework, detailing its components, including the parser, symbolic graph transformations, generator, and optional faithfulness checker. Section §4 presents an application demonstration, focusing on generating hard negative pairs to challenge state-of-the-art text embedding models and reveal potential weaknesses in their linguistic understanding. Finally, Section §5 concludes the paper, discussing the broader implications of our work and potential directions for future research. We release code and data under a public license.[1]

2 Related Work and Background

Method: Semantic graph as an intermediate representation. The value of using semantic graphs as an intermediate representation, particularly AMR, has been highlighted in two recent surveys (Sadeddine et al., 2024; Wein and Opitz, 2024). This approach allows the fusion of neural network power with the expressivity and explicitness of meaning representations, which is especially useful when *interpretability* and *control* are required in an application. Related work applies this principle to style transfer (Jangra et al., 2022), in the MT domain (Wein and Schneider, 2024), and data augmentation (Min et al., 2020; Shou et al., 2022; Shou and Lin, 2023; Ghosh et al., 2024; Kim et al., 2024). Our work generalizes this approach further and imposes an additional check for validating the faithfulness of the generation; Unlike Li et al. (2020) we do not rely on an inflexible rule-parser and/or LLMs for verification, proving the feasibility of efficiently scaling this approach. Notably, the idea of planning/controlling sentence generation through meaning representation dates decades back (i.a., Sondheimer and Nebel, 1986; Mann and Matthiessen, 1983; Kasper, 1989; Wijnen, 1990; Bateman et al., 1990), but was limited by inaccuracies in parsing and generation. With stronger parsing and generation systems now available, we argue that effective usage has become feasible.

Application: Embedding models and benchmarking. Text embedding models are crucial for a wide range of NLP tasks, including semantic search, information retrieval, and NLG evaluation (Clark et al., 2019; Muennighoff, 2022; Gao et al., 2023). Since Reimers and Gurevych (2019)'s foundational "SBERT" work, multiple branches of embedding model research have emerged. These include enhancing model performance through scaling parameters (Wang et al., 2024) or training data (Wang et al., 2022), as well as exploring unsupervised embeddings (Gao et al., 2021) and interpretable embeddings (Opitz and Frank, 2022b). Key questions are: *What is the accuracy of such embeddings?* ***What level of linguistic understanding***

[1] https://github.com/impresso/sentence-smith

resides in those vectors? While the first question is typically addressed through large-scale benchmarks like MTEB (Muennighoff et al., 2023), the second question requires a more fine-grained approach. Moreover, the questions are intertwined, and limits on linguistic understanding might be concealed by averages over large benchmark datasets where individual datasets often have hardly interpretable notions of similarity. We employ SENTENCESMITH and demonstrate its capacity to generate fine-grained evaluation data that test embedding models' ability to assess different linguistic phenomena. With special regard to interpretability of text embeddings (Opitz et al., 2025), our work is also related to a recent/concurrent line of work by Nastase et al. (Nastase and Merlo, 2024; Nastase et al., 2024a,b). In contrast to these works, however, we don't rely on costly human data creation.

Furthermore, the stasis of most benchmarks has also drawn criticism (Fan et al., 2024), and limits evaluation to the available data, with potential ramifications for the trustworthiness of results. By demonstrating how SENTENCESMITH can generate challenging, trustworthy, and interpretable test sets, we pave the way for more customizable, dynamic and interpretable testing of models.

Paraphrases and building minimal pairs. How do two texts relate? In a broader context, our work is related to measuring paraphrases and entailment, both long-standing topics of interest in the NLP and ML (Bhagat and Hovy, 2013; Bowman et al., 2015; Zhou and Bhat, 2021; Opitz et al., 2023; Gubelmann et al., 2023; Krishna et al., 2023). Instead of building or rating paraphrases, we first and foremost build challenging negatives given a paraphrase set, in a controlled manner, such that the relation between negative and paraphrase is transparent (e.g., negation). This may shed also more light on a problem recently highlighted: Notions of paraphrases empirically differ (Michail et al., 2025a) and the differences are not transparent. The controlled manipulations furnished by our SENTENCESMITH framework allow for more in-depth studies. There are also other efforts that note the value of constructing such relation-controlled minimal linguistic pairs (BLiMP, Warstadt et al., 2020; Jumelet et al., 2025)—in contrast to these works, our framework does not require static resources or manual annotation effort.[2]

3 The SENTENCESMITH Framework

3.1 Overview

The formal description of SENTENCESMITH is straightforward. Given s as an input sentence, a parser p and a generator g, SENTENCESMITH conducts a controlled transformation, resulting in a changed sentence s', that is:

$$s' = g \circ t_n \circ ... \circ t_1 \circ p \mid s, \qquad (1)$$

where $\{t_0, ...t_n\}$ are graph transformations that we apply on the graph representation of the input sentence s, output of p. Finally, g generates a text from the final state of the graph.

3.2 Parameterization

Graph model. We rely on Abstract Meaning Representation (AMR, Banarescu et al., 2013) as our graph framework. AMR represents text as directed acyclic graphs, where nodes denote entities and edges capture semantic relations. The overarching goal of AMR is to explicitly encode "Who does what to whom?" This explicitness is a key motivation for our work, as it enables targeted meaning changes within the semantic structure.

Parsing and generation. As is standard in most AMR applications, we employ off-the-shelf parsing and generation models.[3] These models, based on pre-trained BART (Lewis et al., 2020), produce linearized sequence graphs from text (parsing) or text from linearized sequence graphs (generation). To facilitate manipulation, we convert the linearized graph into triples of the form `<u,r,v>`, where `u` and `v` are graph nodes, and `r` is a relation.

Graph transformation. This is the interface where a human (or other controller) can influence the machine-based sentence transformation. Beyond various graph operations, we may wish to, for example, modify truth values by adding a negation to a select predicate, swap semantic roles, or add new information in a controlled manner. Since different use cases of SENTENCESMITH may require distinct transformations, the specific rules applied in our demonstration study are detailed later in §4.

[2] Concurrent work also used LLMs to similarly generate hard negatives from given paraphrases, with similar application scenario (Michail et al., 2025b; Magomere et al., 2025). However, this is a categorically different approach, because no controllable intermediate representation is leveraged, which limits the control over the particular type of relation. Also, the LLM use makes such approaches much more costly, while our approach only depends on smaller neural models.

[3] https://github.com/bjascob/amrlib

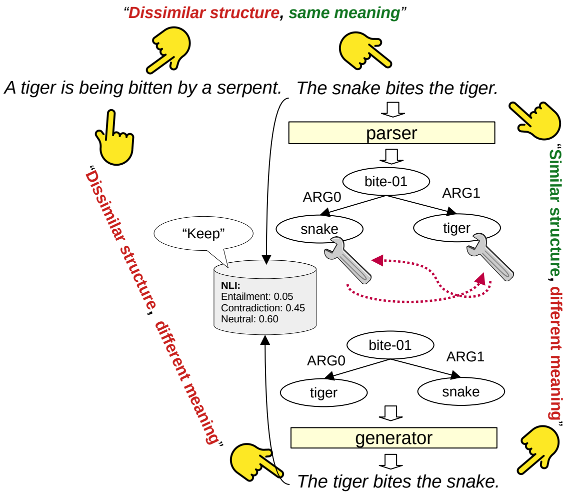
"Dissimilar structure, same meaning"
A tiger is being bitten by a serpent.
The snake bites the tiger.
parser
bite-01
ARG0
ARG1
snake
tiger
"Keep"
NLI:
Entailment: 0.05
Contradiction: 0.45
Neutral: 0.60
bite-01
ARG0
ARG1
tiger
snake
generator
The tiger bites the snake.
"Dissimilar structure, different meaning"
"Similar structure, different meaning"

Figure 2: Breaking the paraphrase relation and creating a challenging test case: The new sentence has a high surface similarity to the input, but it is not a paraphrase. By contrast, the actual paraphrases have a lower superficial overlap, thus posing a challenge to any model, especially those that tend to rely on surface cues.

Validation. Once a transformed sentence is generated, we would like to validate whether the intended transformation was successfully executed. Potential failure cases include noise introduced by the parsing and generation process. We employ a faithfulness check function $check(s, s') \in \{-1, 0, 1\}$, where -1 denotes contradiction between s and s', 0 implies a neutral relation, and 1 indicates that s' is entailed by s.

The criteria for discarding output sentences depend on the application: If the goal is meaning alteration, we should filter out cases where s and s' mutually entail each other. Conversely, if we seek paraphrases, we should discard cases where contradiction or neutrality is detected. SENTENCESMITH parameterizes $check$ with an efficient NLI-based model from Steen et al. (2023). This system is robustness-enhanced through data augmentation and achieves strong results on the TRUE faithfulness benchmark (Honovich et al., 2022).[4]

4 Application Study

As an important application to demonstrate the usefulness of SENTENCESMITH, we focus on fine-grained linguistic testing of embedding models.

Foil idea. The foil concept is illustrated in Figure 2. The objective is to generate a new sentence that is structurally highly similar to an existing one but semantically distinct. If this transformation is applied to one part of a paraphrase pair, the resulting sentence will not maintain a paraphrase relation with either of the original sentences. In the example, we start with the paraphrases a) *A tiger is being bitten by a serpent.* and its paraphrastic variant b) *The snake bites the tiger.* Our parser processes sentence b), and within the generated meaning graph, we perform a *semantic role confusion*, swapping the agent and patient of the *bite* event. Finally, our generator produces c) *The tiger bites the snake*, which is structurally more similar to a) than a) and b) are to each other (i.e., higher lexical overlap). However, the event's semantics have *fundamentally* changed, thereby breaking the paraphrase relation.

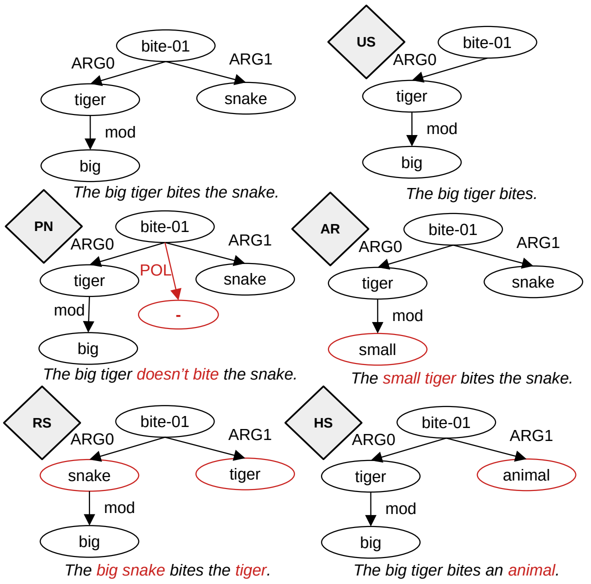

Figure 3: Five aspectual semantic manipulations.

4.1 Setup

4.1.1 Defining Semantic Manipulations

We define five semantic manipulations designed to generate foils from a given sentence: Polarity Negation, Role Swap, Underspecification, Antonym Replacement, and Hypernym Substitution. Examples of all five manipulation types are displayed in Figure 3. Below, we present a more detailed description of each manipulation.

PN: Polarity Negation. This fundamental manipulation assesses embedding models' sensitivity to negation. We define negation as altering the polarity of a predicate within an AMR graph. Specifically, this transformation attaches a `<n, :polarity, ->` edge (in AMR notation, a nega-

[4] We use https://huggingface.co/juliussteen/DeBERTa-v3-FaithAug.

tion) to a randomly selected node n, excluding pronouns. For example, modifying the predicate *approve* by adding this edge results in its opposite, *disapprove* (i.e., in AMR: `<n,:polarity,->` is added to node n, where `<n,:instance,approve>` already exists). The manipulated AMR is then used to generate the negated text. As can be seen in the corresponding example of Figure 3, the truth value of the sentence changes.

RS: Role Swap. This manipulation evaluates how well embedding models distinguish fine-grained semantic relations. It randomly swaps two graph nodes u and v, generalizing the semantic role switch illustrated in Figure 1 and Figure 3, where s (instance of snake) and t (instance of tiger) were swapped.

US: Underspecification. This transformation removes a randomly selected leaf node n from the AMR graph, reducing semantic content, thereby breaking a paraphrase relation. The introduced ambiguity or incompleteness tests how robust embedding models are to partial information.[5] In Figure 3, the patient (a snake) has disappeared.

AR: Antonym Replacement inverts the meaning of a selected node while maintaining the context of the surrounding graph. The system identifies a random non-pronoun node, extracts its lexical stem (e.g., happy in happy-01), and queries WordNet (Miller, 1993) for antonyms. If a suitable antonym is found, the node label is updated while preserving any sense suffix (e.g., -01). This transformation alters meaning more explicitly than polarity negation by directly substituting an opposite term. In the Example for AR given in Figure 3, same as in PN, the truth value changes —while still a *biting* is happening, this time the agent is small rather than big.

HS: Hypernym Substitution. This transformation replaces a word with its hypernym, distorting inferences and breaking paraphrase relationships. Consider *Penguins can't fly*. If *Penguin* is replaced with its hypernym *Bird*, the modified sentence is false ("Birds can't fly"), demonstrating how hypernym substitution alters meaning while preserving surface structure. Following the same WordNet-based approach as in the antonym replacement, non-pronoun node labels are substituted with randomly selected hypernyms. The node's graph position and semantic roles remain unchanged, but abstraction increases, disrupting logical inferences. In Figure 3, we observe how *snake* is replaced with *animal*. This is also some form of underspecification, but speaking through the AMR, it is achieved by replacing (not removing) a concept (which can also cause a change in truth value, as in the aforementioned bird-example).

4.1.2 Data Instantiation

Initial datasets. We use two paraphrase datasets as our base data. The first is PAWS (Yang et al., 2019)[6], from which we take the positive pairs. While PAWS provides structurally similar paraphrases—originally designed to challenge paraphrase detection models with adversarial positive and negative pairs—it is not fully ideal for our purpose. Due to the high structural similarity, we anticipate slightly lower difficulty in the challenge pairs generated from this dataset.

To obtain a more comprehensive benchmark, we additionally leverage a more diverse set of ChatGPT-generated paraphrases (Vorobev and Kuznetsov, 2023)[7], henceforth denoted as GPTP. This dataset consists solely of positive paraphrase pairs. Also unlike PAWS, GPTP exhibits greater structural variability, making our resulting challenge set inherently more difficult. We validate this statistically by computing bag-of-words similarity between paraphrase pairs in each dataset: PAWS yields an average similarity of 0.9, whereas GPTP has a much lower average similarity of 0.4 (see Appendix A.2 for details).

Data induction and filtering. After employing our parser $parse$, a graph transformation $manip$, and the generator gen on every pair, we end up with triples (s, p, f), where s is the original sentence, p a paraphrase, and $f = gen(manip(parse(s)))$ our generated foil (structurally highly similar to s, but of another meaning). For post-processing, to maximize the integrity of the generated benchmark, we apply a harsh quality filtering process through our NLI-based validation module: We only retain those "contradiction" sentence pairs with a semantic relationship confidence score greater than 90%.

[5] Note that it can also introduce a polarity change, in case a negation node in the AMR is removed. The notion of "Underspecification" refers to the AMR structure being reduced.

[6] https://huggingface.co/datasets/google-research-datasets/paws-x/tree/main

[7] https://huggingface.co/datasets/humarin/chatgpt-paraphrases

	total	PN %	RS %	US %	AR %	HS %
PAWS	1,737	39.5	18.5	9.7	27.2	5.0
GPTP	11,456	36.5	15.3	9.6	30.4	8.2

Table 1: Final data set statistics.

	Meaning accuracy		Fluency					
			worse		same		better	
	A1	A2	A1	A2	A1	A2	A1	A2
PAWS	98	99	34	41	30	33	36	26
GPTP	100	99	42	45	35	37	23	18

Table 2: Manual assessment of the generation quality. All numbers are %s. A1 and A2 are human annotators.

This ensures that the final dataset focuses on maximally validated paraphrase foils. However, we will later also experiment with a different filtering criterion. Final dataset statistics, including distribution of transformations, are shown in Table 1.

Human evaluation of generation. We conduct a human annotation to assess the quality of our generations. Specifically, we are interested in two variables. First, *Does the meaning of sentence B differ from that of sentence A (no, or yes)?* This lets us assess how faithfully the final output of SENTENCESMITH has achieved our main goal: The manipulation of text meaning. Second, we wonder about the fluency of the final generation: *Given sentence A, is the fluency of sentence B worse, about the same, or improved?* We task two annotators to annotate 100 randomly sampled pairs from each of PAWS and GPTP, so 200 in total.

Table 2 presents the results. Our primary goal—meaning manipulation—is achieved to an overwhelming degree, with only one or two exceptions across both samples. While fluency tends to degrade on average, a notable portion of outputs remain equally fluent or even improve in fluency.

Examples of our generated foils are shown in Appendix A.3, Table 5. E.g., through adding a negative polarity to the node "need", the statement *But you need to get somebody like Warren to do it* becomes *But you don't need to get somebody like Warren to do it*, which is superficially highly similar, but has the opposite meaning of the original paraphrase *You should find someone similar to Warren to handle it* that is structurally much more different on the surface.

4.2 Evaluating Embedding Models

4.2.1 Embedding Model Selection

With our benchmark fully set up, we evaluate 29 off-the-shelf text embedding models. The selection aims to balance performance and efficiency, comprising top-ranked models at the time of writing as well as other widely used baselines. These include LaBSE (Feng et al., 2022), SBERT (Reimers and Gurevych, 2019), Jina (Günther et al., 2023), and E5 (Wang et al., 2022). Additional models were drawn from the MTEB leaderboard[8] to provide broader coverage of model types.

It is important to note that names like "SBERT" or "E5" represent techniques rather than specific model instances. Therefore, from here on, we reference specific models by their Hugging Face identifiers. For instance, `all-mpnet-base-v2` refers to a widely used embedding model that applies the SBERT training methodology to a self-supervised pre-trained MPNET model (Song et al., 2020).

4.2.2 Evaluation Measures

Consider any model $\mathcal{E}$ that maps a pair of texts to a real-valued "similarity score" (in our case we have text embedding models that construct the embeddings, and the cosine similarity builds the similarity). To assess the quality of such a model on our induced benchmark, we compute two key evaluation metrics. The first is **triplet accuracy** ($TACC$), a simple and interpretable classification-oriented score that measures the ratio of cases where the embedding model remains robust to the foil. We consider a set with triplets $T = \{(s_i, p_i, f_i)\}_{i=1}^{n}$, where s_i is a text, p_i its paraphrase, and f_i our generated foil designed to mislead the model into assigning it a higher similarity score. Then

$$TACC = \frac{1}{|T|} \sum_{(s,p,f)\in T} \mathcal{I}[\mathcal{E}(s,p) > \mathcal{E}(s,f)],$$

where $\mathcal{I}[c]$ returns 1 if the condition c is true, and 0 else. As a more standard metric that is commonly used to evaluate tasks where a floating value (here: similarity) must be compared against a discrete label (here: paraphrase/not-paraphrase), we also measure the **Area Under the Receiver Operator Curve** (AUC) that represents a softer score directly based on the real valued output scores of embedding models. For this, we dichotomize the

[8] `https://huggingface.co/spaces/mteb/leaderboard`

dataset into positive pairs $\{(s_i, p_i)\}_{i=1}^n$ and negative pairs $\{(s_i, f_i)\}_{i=1}^n$, essentially inducing a binary paraphrase classification task. Since it is hard to say which of the two metrics is generally more informative, models should clearly excel in both and thus (Opitz, 2024, Rec. 3 in Section 8), to compute a single "performance" number for a given dataset, we are using an harmonic mean of TACC and AUC.

4.3 Results

Model Name	**PAWS**	**GPTP**	**AVG**
`sentence-t5-large`	0.9506	0.8026	0.8704
`ember-v1`	0.9562	0.7513	0.8415
`bge-base-en-v1.5`	0.9506	0.6910	0.8003
`e5-base-v2`	0.9378	0.6930	0.7970
`GIST-Embedding-v0`	0.9297	0.6954	0.7957
`FAB-Ramy-v1`	0.9190	0.6928	0.7900
`gte-base-en-v1.5`	0.9062	0.6947	0.7865
`all-mpnet-base-v2`	0.9046	0.6920	0.7841
`instructor-base`	0.9242	0.6503	0.7634
`paraphrase-MiniLM-L12-v2`	0.9509	0.6286	0.7569
`nomic-embed-text-v1.5`	0.8896	0.6519	0.7524
`jina-embeddings-v2-base-en`	0.9314	0.5965	0.7272
`MedEmbed-small-v0.1`	0.9237	0.5980	0.7260
`stella-base-en-v2`	0.9508	0.5724	0.7146
`Wartortle`	0.9556	0.5639	0.7093
`LaBSE`	0.9657	0.5444	0.6963
`all-MiniLM-L12-v2`	0.9234	0.5586	0.6961
`gtr-t5-large`	0.8501	0.5762	0.6869
`cde-small-v1`	0.8507	0.5653	0.6792
`gte-micro`	0.9320	0.5315	0.6769
`msmarco-bert-co-condensor`	0.8724	0.5474	0.6727
`contriever-base-msmarco`	0.8880	0.5214	0.6570
`snowflake`	0.8852	0.5104	0.6475
`Ivysaur`	0.9194	0.4889	0.6384
`Venusaur`	0.9323	0.4459	0.6033
`distiluse-base-v2`	0.9475	0.4393	0.6003
`allenai-specter`	0.8143	0.3908	0.5281
`SGPT-125M`	0.7491	0.3982	0.5200
`komninos`	0.8905	0.3431	0.4953

Table 3: Main results. Shown numbers are harmonic means of AUC and TACC measures. AVG is their arithmetic mean across datasets.

Table 3 shows the main results. The top-ranked model is `sentence-t5-large`, showing an AVG score of 0.87, outperforming the worst-ranked model `SGPT-125M` by about 30 percentage points (pp.). On PAWS, several models show strong performance of more than 0.90. This is likely due to the higher initial structural similarity of the paraphrases (for statistics, see the Appendix, §A.4), reducing the effectiveness of some foils. On GPTP, with its more varied structural differences, the differences between models grows, and even the best performing model only barely exceeds an AVG of 0.8. For individual metrics on each dataset, see Appendix A.4: Table 6 shows evaluation on GPTP, and Table 7 on PAWS.

4.4 Analysis

A fine-grained picture is shown in Figure 4. For each embedding model, we plot the $TACC$ accuracy across different manipulation types within the GPTP dataset. We make several interesting observations: 1. `sentence-t5-large` excels in all categories. 2. Many embedding models are moderately robust to polarity changes (purple and red bars), an important property for models of which we expect finer linguistic understanding. However, this is not consistent across models, with some exhibiting a $TACC$ accuracies at or below 40%. Notably, the (less recent) `all-mpnet-base-v2` shows the most balanced performance across all manipulations, suggesting its suitability for use cases requiring uniform sensitivity to diverse linguistic manipulations. On average (AVG, right column), role swap is the most difficult transformation, while hypernym substitution is the easiest, though the differences are not pronounced. Overall, the results indicate considerable potential for improvement in embedding models' fine-grained linguistic understanding.

NLI filtering ablation study. To ensure the highest faithfulness of our automatically induced challenge cases, we selected only pairs where our NLI-based validation assigned a high probability of contradiction. In this experiment, we construct an alternative dataset containing more ambiguous cases by selecting pairs labeled as *neutral* with probabilities between 50% and 80%. Specifically, we examine how much the final ranking deviates between the two datasets.

The results are shown in Table 8, Appendix A.4. Interestingly, `sentence-t5-large`, consistent with all previous experiments, again achieves the top rank, even though the intersection of the two datasets is zero. Overall, the ranking stays mostly stable (Spearman's rank correlation coefficient: 0.91), with no substantial changes.

Comparison against *MTEB* ranking. Our created testing challenge focuses on *linguistic understanding* and can be dynamically generated and adapted. On the other hand, static and large benchmarks like MTEB cover a large spectrum of *tasks*, including STS, but also IR, argument mining, and

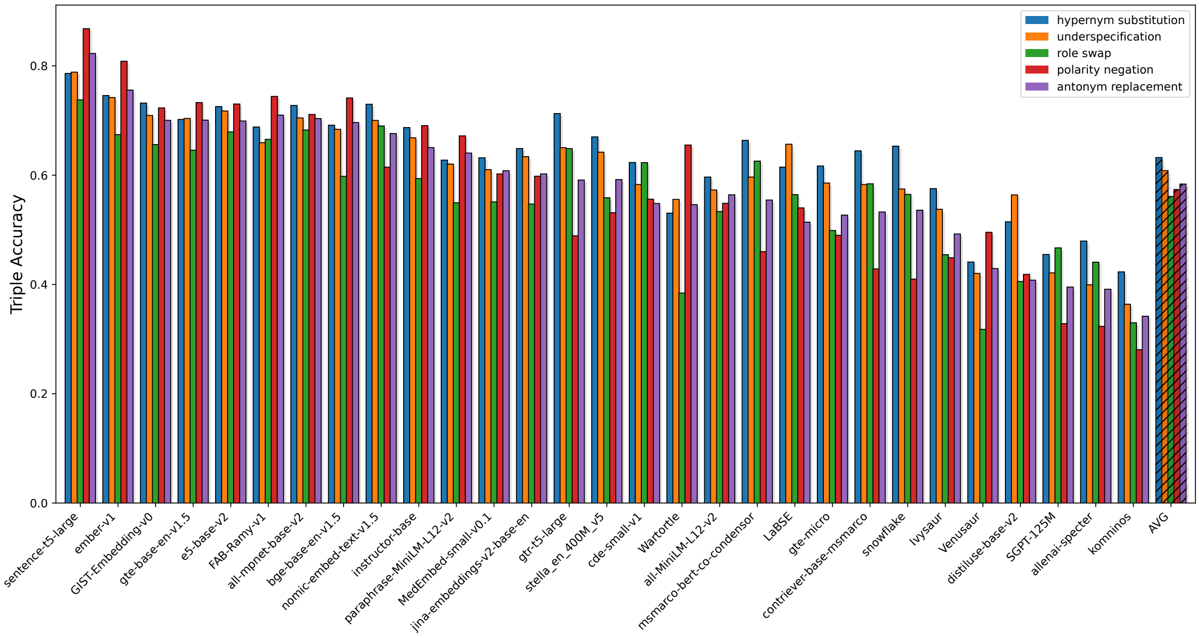

Figure 4: Fine-grained linguistic benchmarking of text embedding models.

so on. Therefore, we might expect a different ranking of models. Models that would score high in our benchmark but lower on MTEB could point at unleveraged linguistic strengths of a model that might not be fully reflected by MTEB. On the other hand, a relatively lower ranking in our challenge may indicate a model weakness and suggest vulnerabilities in difficult cases.

In our Appendix, Table 9, we compare the ranking of our models against the relative rankings on MTEB. Notably, the relatively best ranked model on MTEB (`cde-small-v1`, as of October 1st, 2024, the best model of a size smaller than 400M params) only reaches place 20 in our challenge, suggesting poorer semantic sentence understanding than `sentence-t5-large`, a model of similar size based on a seq-to-seq T5 transformer. Thus, our results suggest potential vulnerabilities of `cde-small-v1`, especially towards negation and polarity. To assess the overall difference between our ranking versus the MTEB ranking, we again calculate the Spearman's rank correlation. We receive a score of 0.54, suggesting a significant but only moderate alignment of the two benchmarks. While MTEB contains a large number of datasets, constituting a very large overall benchmark, our datasets have the advantage that they contain *fresh, unseen benchmarking data* that can be dynamically generated, and that allow for targeted probing of models' *linguistic understanding*.

Metric	*LOWER*	E5-MISTRAL	GTE-QWEN	*UPPER*
TACC	0.326	0.416	0.540	0.820

Table 4: Results for embeddings from very large models on GPTP dataset. *LOWER* and *UPPER* are reference values from the other models tested on GPTP. The worst performing being `komnios` and the best performing `sentence-t5-large`.

Embeddings from very large models. We additionally use our data to probe sentence understanding of text embeddings from two very large language models: E5 with MISTRAL LLM 7B as backbone (Wang et al., 2022; Jiang et al., 2023), and GTE with QWEN 7B LLM as backbone (Li et al., 2023; Bai et al., 2023). Interestingly, the resulting scores (Table 4) are surprisingly low, placing the models in the lower-middle rankings.[9]. In contrast, smaller models, specifically T5-based embeddings, perform better. Our hypothesis is that, since these LLM-based models are optimized for retrieval and longer-context tasks, their embeddings may be less sensitive to fine-grained aspects of sentence meaning.

5 Conclusion

Our SENTENCESMITH framework facilitates controlled and highly transparent meaning transformations of text. We showcased SENTENCESMITH's

[9] The weakest aspect is polarity, where both models get more than 50% of cases wrong.

utility in an NLP task: Dynamic benchmarking and analysis of text embedding models. Concretely, we used it to produce challenging foils from paraphrase pairs, carefully breaking the paraphrase relation while a high degree of superficial similarity. Through this, we shed light on the robustness of text embedding models and their ability to distinguish fine-grained linguistic phenomena.

Limitations

We note two main types of limitations: those that relate to the model of meaning representation, and those that affect conclusions drawn from our automatically induced benchmark.

Meaning Representation Model. It is important to note that SENTENCESMITH directly scales with improvements in the area of meaning representations. While AMR has the great advantage of large data sets as well as reasonably robust parsers and generators, it also comes with limitations and drawbacks that have been, over the years, carefully outlined by research. There is the mono-linguality (Banarescu et al., 2013), then there may be meaning non-isomorphisms due to ambiguity (Wein, 2025), and lack of some scope (Pustejovsky et al., 2019) and tense aspects (Donatelli et al., 2018). Moreover, while AMR parsers and generators seem to score high on benchmarks (e.g., Vasylenko et al., 2023), these tasks remain far from solved (Manning et al., 2020; Opitz and Frank, 2022a; Yang and Schneider, 2024), as is also supported by our output investigation where we observed some fluency issues, that may both be due to generation or parsing issues. This setup restricts our framework to English, because robust parsers and generators are not yet widely available for other languages. However, in the context of multilinguality, promising advances in cross-lingual parsing have emerged through "Universal Representations" (Van Gysel et al., 2021). Exploiting the wealth of AMR tools, we might also effectively use an off-the-shelf MT model to wrap the English AMR parsing and generation process (Uhrig et al., 2021), or use cross-lingual AMR parsing for certain languages (Vanroy and Van de Cruys, 2024; Kang et al., 2024). Given sufficient quality of the MT system, this simple approach could already work for some cross-lingual use-cases of our SENTENCESMITH.

On drawing conclusions about embedding model performance. Embedding models are widely used in research and industry, making it important to investigate differences in their performance. Our application study and its induced dataset *do* allow us to test models' sensitivity to paraphrases while also highlighting the specific linguistic phenomena that drive this sensitivity. However, different embedding objectives and downstream settings may prioritize relevance as the primary goal, in which case this testing strategy may not provide the most informative signal for model selection. The ranking of models in our setting is conditional on their "similarity" scores serving as proxies for semantic similarity, rather than on a neutral or universal evaluation of embedding quality. Thus, the results should not be misinterpreted as a general quality difference between embedding models. Instead, they reflect the constraint that performance (or, more cautiously, behavior) is evaluated only with respect to specific linguistic paraphrase phenomena. A strength of this targeted evaluation, however, is that it enables us to diagnose model weaknesses in particular categories. Future work may adapt the manipulation rules in ways so that additional objectives can be effectively evaluated.

Acknowledgments

We thank the anonymous reviewers for their comments, and the meta-reviewer for their detailed feedback. Three authors received funding through the project *Impresso – Media Monitoring of the Past II. Beyond Borders: Connecting Historical Newspapers and Radio*. Impresso is a research project funded by the Swiss National Science Foundation (SNSF 213585) and the Luxembourg National Research Fund (17498891).

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

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, Sofia, Bulgaria. Association for Computational Linguistics.

John Bateman, Robert Kasper, Jörg Schütz, and Erich Steiner. 1990. Interfacing an english text generator with a german MT analysis. In *Interaktion und Kommunikation mit dem Computer: Jahrestagung der Gesellschaft für Linguistische Datenverarbeitung (GLDV). Ulm, 8.-10. März 1989 Proceedings*, pages 155–163. Springer.

Rahul Bhagat and Eduard Hovy. 2013. Squibs: What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Igor A Bolshakov and Alexander Gelbukh. 2004. Synonymous paraphrasing using wordnet and internet. In *International conference on application of natural language to information systems*, pages 312–323. Springer.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760, Florence, Italy. Association for Computational Linguistics.

Lucia Donatelli, Michael Regan, William Croft, and Nathan Schneider. 2018. Annotation of tense and aspect semantics for sentential AMR. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 96–108, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Lizhou Fan, Wenyue Hua, Lingyao Li, Haoyang Ling, and Yongfeng Zhang. 2024. NPHardEval: Dynamic benchmark on reasoning ability of large language models via complexity classes. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4092–4114, Bangkok, Thailand. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Sreyan Ghosh, Utkarsh Tyagi, Sonal Kumar, Chandra Kiran Evuru, Ramaneswaran S, S Sakshi, and Dinesh Manocha. 2024. ABEX: Data augmentation for low-resource NLU via expanding abstract descriptions. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 726–748, Bangkok, Thailand. Association for Computational Linguistics.

Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. 2024. AI control: Improving safety despite intentional subversion. In *Forty-first International Conference on Machine Learning*.

Reto Gubelmann, Aikaterini-lida Kalouli, Christina Niklaus, and Siegfried Handschuh. 2023. When truth matters - addressing pragmatic categories in natural language inference (NLI) by large language models (LLMs). In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 24–39, Toronto, Canada. Association for Computational Linguistics.

Michael Günther, Louis Milliken, Jonathan Geuter, Georgios Mastrapas, Bo Wang, and Han Xiao. 2023. Jina embeddings: A novel set of high-performance sentence embedding models. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 8–18, Singapore. Association for Computational Linguistics.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Kuo-Chuan Huang, James Geller, Michael Halper, Yehoshua Perl, and Junchuan Xu. 2009. Using wordnet synonym substitution to enhance UMLS source integration. *Artificial Intelligence in Medicine*, 46(2):97–109.

Anubhav Jangra, Preksha Nema, and Aravindan Raghuveer. 2022. T-STAR: Truthful style transfer using AMR graph as intermediate representation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8805–8825, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Jaap Jumelet, Leonie Weissweiler, Joakim Nivre, and Arianna Bisazza. 2025. MultiBLiMP 1.0: A massively multilingual benchmark of linguistic minimal pairs. *arXiv preprint arXiv:2504.02768*.

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.

Robert T. Kasper. 1989. A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Minji Kim, Whanhee Cho, Soohyeong Kim, and Yong Suk Choi. 2024. Simple data transformations for mitigating the syntactic similarity to improve sentence embeddings at supervised contrastive learning. *Adv. Intell. Syst.*, 6(8).

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems*, volume 36, pages 27469–27500. Curran Associates, Inc.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chuanrong Li, Lin Shengshuo, Zeyu Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020. Linguistically-informed transformations (LIT): A method for automatically generating contrast sets. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 126–135, Online. Association for Computational Linguistics.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Jabez Magomere, Emanuele La Malfa, Manuel Tonneau, Ashkan Kazemi, and Scott A. Hale. 2025. When claims evolve: Evaluating and enhancing the robustness of embedding models against misinformation edits. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22374–22404, Vienna, Austria. Association for Computational Linguistics.

William C. Mann and Christian M. I. M. Matthiessen. 1983. Nigel: A systemic grammar for text generation. Technical Report (ISI/RR-83-105) ISI/RR-83-105, Information Sciences Institute, University of Southern California, Marina del Rey, CA, USA.

Emma Manning, Shira Wein, and Nathan Schneider. 2020. A human evaluation of AMR-to-English generation systems. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4773–4786, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Andrianos Michail, Simon Clematide, and Juri Opitz. 2025a. PARAPHRASUS: A comprehensive benchmark for evaluating paraphrase detection models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8749–8762, Abu Dhabi, UAE. Association for Computational Linguistics.

Andrianos Michail, Simon Clematide, and Rico Sennrich. 2025b. Examining multilingual embedding models cross-lingually through llm-generated adversarial examples. *Preprint*, arXiv:2502.08638.

George A. Miller. 1993. WORDNET: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Vivi Nastase and Paola Merlo. 2024. Tracking linguistic information in transformer-based sentence embeddings through targeted sparsification. In *Proceedings of the 9th Workshop on Representation Learning for NLP (RepL4NLP-2024)*, pages 203–214, Bangkok, Thailand. Association for Computational Linguistics.

Vivi Nastase, Giuseppe Samo, Chunyang Jiang, and Paola Merlo. 2024a. Exploring Italian sentence embeddings properties through multi-tasking. In *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, pages 620–630, Pisa, Italy. CEUR Workshop Proceedings.

Vivi Nastase, Giuseppe Samo, Chunyang Jiang, and Paola Merlo. 2024b. Exploring syntactic information in sentence embeddings through multilingual subject-verb agreement. In *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, pages 631–643, Pisa, Italy. CEUR Workshop Proceedings.

Juri Opitz. 2024. A closer look at classification evaluation metrics and a critical reflection of common evaluation practice. *Transactions of the Association for Computational Linguistics*, 12:820–836.

Juri Opitz and Anette Frank. 2022a. Better Smatch=better parser? AMR evaluation is not so simple anymore. In *Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems*, pages 32–43, Online. Association for Computational Linguistics.

Juri Opitz and Anette Frank. 2022b. SBERT studies meaning representations: Decomposing sentence embeddings into explainable semantic features. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 625–638, Online only. Association for Computational Linguistics.

Juri Opitz, Lucas Möller, Andrianos Michail, Sebastian Padó, and Simon Clematide. 2025. Interpretable text embeddings and text similarity explanation: A survey. *EMNLP 2025*.

Juri Opitz, Shira Wein, Julius Steen, Anette Frank, and Nathan Schneider. 2023. AMR4NLI: Interpretable and robust NLI measures from semantic graphs. In *Proceedings of the 15th International Conference on Computational Semantics*, pages 275–283, Nancy, France. Association for Computational Linguistics.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. LogicBench: Towards Systematic Evaluation of Logical Reasoning Ability of Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13679–13707, Bangkok, Thailand. Association for Computational Linguistics.

James Pustejovsky, Ken Lai, and Nianwen Xue. 2019. Modeling quantification and scope in Abstract Meaning Representations. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 28–33, Florence, Italy. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Zacchary Sadeddine, Juri Opitz, and Fabian Suchanek. 2024. A survey of meaning representations – from theory to practical utility. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2877–2892, Mexico City, Mexico. Association for Computational Linguistics.

Ziyi Shou, Yuxin Jiang, and Fangzhen Lin. 2022. AMR-DA: Data augmentation by Abstract Meaning Representation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3082–3098, Dublin, Ireland. Association for Computational Linguistics.

Ziyi Shou and Fangzhen Lin. 2023. Evaluate AMR graph similarity via self-supervised learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16112–16123, Toronto, Canada. Association for Computational Linguistics.

Norman K. Sondheimer and Bernhard Nebel. 1986. A logical-form and knowledge-base design for natural language generation. In *Strategic Computing - Natural Language Workshop: Proceedings of a Workshop Held at Marina del Rey, California, May 1-2, 1986*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Julius Steen, Juri Opitz, Anette Frank, and Katja Markert. 2023. With a little push, NLI models can robustly and efficiently predict faithfulness. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 914–924, Toronto, Canada. Association for Computational Linguistics.

Sarah Uhrig, Yoalli Garcia, Juri Opitz, and Anette Frank. 2021. Translate, then parse! a strong baseline for cross-lingual AMR parsing. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 58–64, Online. Association for Computational Linguistics.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

Bram Vanroy and Tim Van de Cruys. 2024. Less is enough: Less-Resourced multilingual AMR parsing. In *Proceedings of the 20th Joint ACL - ISO Workshop on Interoperable Semantic Annotation @ LREC-COLING 2024*, pages 82–92, Torino, Italia. ELRA and ICCL.

Pavlo Vasylenko, Pere Lluís Huguet Cabot, Abelardo Carlos Martínez Lorenzo, and Roberto Navigli. 2023. Incorporating graph information in transformer-based AMR parsing. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1995–2011, Toronto, Canada. Association for Computational Linguistics.

Vladimir Vorobev and Maxim Kuznetsov. 2023. ChatGPT paraphrases dataset.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916, Bangkok, Thailand. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Shira Wein. 2025. Ambiguity and disagreement in Abstract Meaning Representation. In *Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation*, pages 145–154, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Shira Wein and Juri Opitz. 2024. A survey of AMR applications. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6856–6875, Miami, Florida, USA. Association for Computational Linguistics.

Shira Wein and Nathan Schneider. 2024. Lost in translationese? reducing translation effect using Abstract Meaning Representation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–765, St. Julian's, Malta. Association for Computational Linguistics.

Frank Wijnen. 1990. The development of sentence planning. *Journal of Child Language*, 17(3):651–675.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2025. Are large language models really good logical reasoners? a comprehensive evaluation and beyond. *IEEE Trans. Knowl. Data Eng.*, 37(4):1620–1634.

Xiulin Yang and Nathan Schneider. 2024. The relative clauses AMR parsers hate most. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 151–161, Torino, Italia. ELRA and ICCL.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. 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 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Jianing Zhou and Suma Bhat. 2021. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Appendix

A.1 Example of Data Generation Process

A running example of a complete data generation and selection process with SENTENCESMITH is shown in Figure 5.

PN: the original sentence *I'm happy that things are going so well* transforms into *I'm happy things aren't going so well for me*. Here, we added `<w,:polarity,->` to node `w`, where `<w,:instance,well-09>` represents "well-being." While the resulting text may sound counterintuitive ("Happy things are not going well"), the goal is successfully achieved: the meaning is strictly altered (paraphrase relation is broken), while the surface structure remains highly similar with many overlapping tokens.

RS: In our example, *I'm happy that things are going so well* becomes *So happy things are going so well for me*. In the AMR graph, `i` (instance of "I") and `s` (instance of "so") were swapped. While the original sentence implies an unspecified referent (*who* is it that things are going well for?), the transformed sentence explicitly assigns this role to an entity ("me") and also shifts emphasis ("so happy" vs. "so well"), altering meaning while preserving structural similarity.

US: *I'm happy that things are going so well* is transformed into *Happy things are going so*

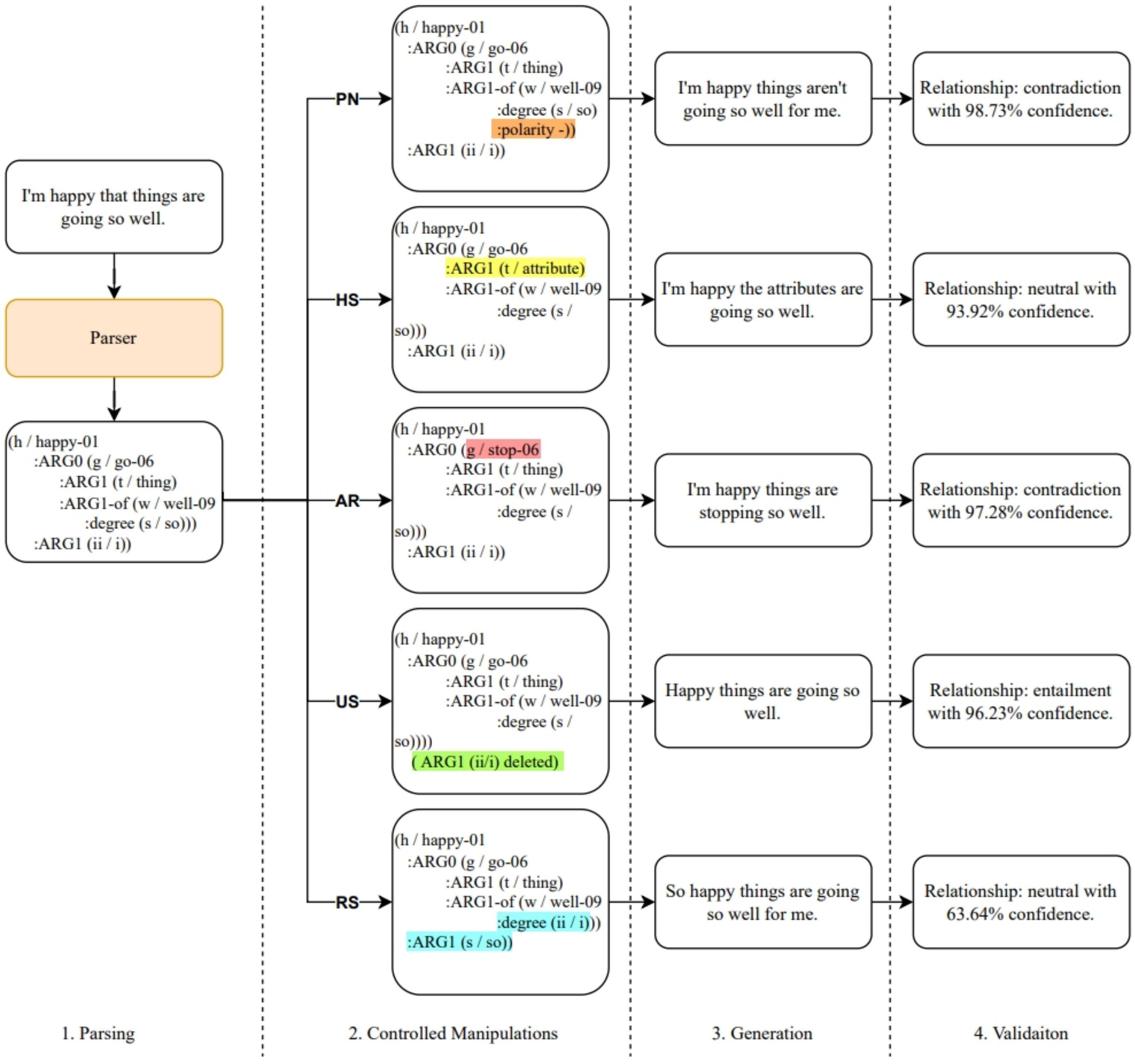
I'm happy that things are going so well.
Parser
(h / happy-01 :ARG0 (g / go-06 :ARG1 (t / thing) :ARG1-of (w / well-09 :degree (s / so))) :ARG1 (ii / i))
PN
HS
AR
US
RS
(h / happy-01 :ARG0 (g / go-06 :ARG1 (t / thing) :ARG1-of (w / well-09 :degree (s / so) :polarity -)) :ARG1 (ii / i))
(h / happy-01 :ARG0 (g / go-06 :ARG1 (t / attribute) :ARG1-of (w / well-09 :degree (s / so))) :ARG1 (ii / i))
(h / happy-01 :ARG0 (g / stop-06 :ARG1 (t / thing) :ARG1-of (w / well-09 :degree (s / so))) :ARG1 (ii / i))
(h / happy-01 :ARG0 (g / go-06 :ARG1 (t / thing) :ARG1-of (w / well-09 :degree (s / so)))) (ARG1 (ii/i) deleted)
(h / happy-01 :ARG0 (g / go-06 :ARG1 (t / thing) :ARG1-of (w / well-09 :degree (ii / i))) :ARG1 (s / so))
I'm happy things aren't going so well for me.
I'm happy the attributes are going so well.
I'm happy things are stopping so well.
Happy things are going so well.
So happy things are going so well for me.
Relationship: contradiction with 98.73% confidence.
Relationship: neutral with 93.92% confidence.
Relationship: contradiction with 97.28% confidence.
Relationship: entailment with 96.23% confidence.
Relationship: neutral with 63.64% confidence.
1. Parsing
2. Controlled Manipulations
3. Generation
4. Validaiton

Figure 5: Running example of the full data generation process with SENTENCESMITH. Under the strict quality filtering criterion of $>90\%$ contradiction, only PN and AR are retained from the five candidate generations. In the ablation study with the alternative, less stringent neutral filter, only the RS is retained.

well. Here, the first-person subject ("I") is removed, which could also lead to an alternative interpretation: that things themselves are happy. Arguably, this change could still be considered a paraphrase in some pragmatic contexts, and we would thus like to filter it out from our final data. This is exactly where the NLI validation module proves valuable—it assigns an entailment probability of 96.2%, indicating that this particular instance might not be suitable as a foil.

AR: *I'm happy that things are going so well* is transformed into *I'm happy things are stopping so well*, where the predicate go-01 is replaced with stop-01 in the AMR graph, that is, `<g,:instance,go-01>` becomes `<g,:instance,stop-01>`. This results in a fundamental semantic shift: the original sentence conveys progression (things are going), while the transformed (slightly weird) sentence implies cessation (things are stopping), disrupting the paraphrase relation but maintains high lexical overlap.

HS: *I'm happy that things are going so well* becomes*I'm happy the attributes are going so well.* Here,"thing", an already quite abstract term, is replaced with an even more abstract alternative, "attribute", leading to a grammatically valid but semantically odd sentence. In AMR, this corresponds to the change `<t,:instance,thing>` → `<t,:instance,attribute>`, demonstrating how even subtle lexical shifts can significantly impact meaning. The NLI validator assigns it a "neutral"-label, with very high confidence. Thus, if wished for high strictness, this concrete example can also be filtered out.

A.2 Bag-of-words Divergence Analysis

To further investigate the semantic divergence between the original sentences and their paraphrases in both datasets, we employ a simple bag-of-words approach. Specifically, we convert each sentence to lowercase, remove punctuation, split whitespace, optionally filter out English stopwords, and treat the remaining tokens as sets. We compute the *Jaccard similarity* between the original and paraphrased token sets, defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|},$$

and use $1 - J(A, B)$ as a simple measure of divergence. We compare two datasets: one from ChatGPT and one from PAWS. For the ChatGPT dataset, which consists of 11,457 sentence-paraphrase pairs, the average Jaccard similarity is approximately 0.3971. In contrast, the PAWS dataset (1,738 pairs) exhibits a much higher average Jaccard similarity of about 0.9002, indicating a much smaller structural divergence that leads to less challenging foils. This indicates that the ChatGPT paraphrases are, on average, much more lexically varied from the original sentences than those in PAWS. From a practical standpoint, this suggests that the ChatGPT-based paraphrasing process produces more diverse rewordings, potentially offering broader coverage for downstream tasks.

A.3 Examples of Manipulations

Examples of some of our generated foils are shown in Table 5.

A.4 Additional Tables

Table 6 presents embedding model rankings on the GPTP dataset, whereas Table 7 reports rankings on the PAWS dataset. These two tables therefore correspond to different underlying data sources. In contrast, Table 8 provides additional results for GPTP only, obtained under an alternative quality filtering criterion in our NLI validator that retains more subtle examples with a probability between 50% and 80% of being labeled neutral.

Text	Choices	Type
The majority of Havocs served with the Soviets, but the US and Great Britain also used the planes.	**T:** Most Havocs were utilized by the Soviets, although the US and Great Britain also employed them. **F:** Havocs also serve with the Soviet Union but the majority of their use is by the US and Britain.	RS
Universities minister David Willetts said all universities can do is ask students if they have booked a flight home.	**T:** According to David Willetts, universities can only inquire about students' flight reservations for their return home. **F:** University Minister David Blankts said the university could no longer do the asking whether students had booked flights home.	AR
It was known as a relatively easy plane to fly with good handling during takeoff and landing.	**T:** The aircraft was recognized for its ease of operation and smooth handling during takeoff and landing. **F:** It is known that it is not a relatively easy flying plane that is well-handled when taking off and landing.	PN
The reason for the crash is not known.	**T:** The cause of the accident is uncertain. **F:** The reason for the crash was known.	US
Carcasses were also found near the contaminated watering holes.	**T:** The polluted watering holes were also discovered to have nearby carcasses. **F:** The carcass was also found far from the contaminated water hole.	AR
The auction house also sold one of the two gun belts owned by Jesse James at the time of his death.	**T:** One of Jesse James' two gun belts at the time of his death was also sold by the auction house. **F:** The auction house also gave up one of Jesse James's two gun belts when he died.	HS
Jean-Marc Wenger, who lives in Klingau, found the gold.	**T:** The gold was discovered by Jean-Marc Wenger, a resident of Klingau. **F:** Jean-Marc Klonau a Wenger resident found gold.	RS
Jean-Marc Wenger, who lives in Klingau, found the gold.	**T:** The gold was discovered by Jean-Marc Wenger, a resident of Klingau. **F:** Jean-Marc Wenger who lives in Klanau has lost gold.	AR
But at the moment it is a complete mystery.	**T:** Currently, it remains an enigma. **F:** But it is not a complete mystery at the moment.	PN
Thibaut Courtois was out quickly to thwart Sterling as Liverpool looked to get back in the game.	**T:** Liverpool attempted to make a comeback, but Thibaut Courtois swiftly prevented Sterling from scoring. **F:** When Liverpool looked to get back in the game Thibaut Sterling was quick out to thwart Courttois.	RS
Willis visited the Neon Museum in 2013 to celebrate her 90th birthday.	**T:** In 2013, Willis marked her 90th birthday by visiting the Neon Museum. **F:** In 2013 to celebrate his 90th birthday he visited the Neo Museum.	US
At least 3,000 Brussels bureaucrats earn more than David Cameron, it emerged yesterday.	**T:** Yesterday, it was revealed that over 3,000 Brussels bureaucrats earn a higher salary than David Cameron. **F:** It emerged yesterday that at least 3000 bureaucrats in Brussels earn less than David Cameron.	AR
We were going in water until hit the hill and spun.	**T:** We were traveling through water until we hit the hill and spun. **F:** We went up the hill until the water hit and we spun.	RS
But you need to get somebody like Warren to do it.	**T:** You should find someone similar to Warren to handle it. **F:** But you don't need to get somebody like Warren to do it.	PN
Julian E. Zelizer says Democrats should be questioning themselves on several key points.	**T:** Democrats ought to be reflecting on various crucial aspects, according to Julian E. Zelizer. **F:** Julian E. Zelizer said Democrats should question him on several key points.	RS

Table 5: Example cases from our automatically induced challenge benchmark. T: The actual paraphrase. F: A generated foil, close to the input text in surface form, but different in meaning.

Model Name	TACC	AUC	AVG
sentence-t5-large	0.8197	0.7862	0.8026
ember-v1	0.7600	0.7427	0.7513
GIST-Embedding-v0	0.7050	0.6860	0.6954
gte-base-en-v1.5	0.7041	0.6856	0.6947
e5-base-v2	0.7111	0.6758	0.6930
FAB-Ramy-v1	0.7086	0.6776	0.6928
all-mpnet-base-v2	0.7049	0.6795	0.6920
bge-base-en-v1.5	0.6957	0.6864	0.6910
nomic-embed-text-v1.5	0.6622	0.6420	0.6519
instructor-base	0.6609	0.6401	0.6503
paraphrase-MiniLM-v2	0.6347	0.6227	0.6286
MedEmbed-small-v0.1	0.5991	0.5969	0.5980
jina-embeddings-v2-base-en	0.5988	0.5943	0.5965
gtr-t5-large	0.5780	0.5744	0.5762
stella-base-en-v2	0.5756	0.5693	0.5724
cde-small-v1	0.5717	0.5590	0.5653
Wartortle	0.5604	0.5674	0.5639
all-MiniLM-L12-v2	0.5570	0.5603	0.5586
msmarco-bert-co-condensor	0.5437	0.5511	0.5474
LaBSE	0.5529	0.5362	0.5444
gte-micro	0.5257	0.5375	0.5315
contriever-base-msmarco	0.5163	0.5265	0.5214
snowflake	0.5074	0.5135	0.5104
Ivysaur	0.4815	0.4966	0.4889
Venusaur	0.4362	0.4561	0.4459
distiluse-base-v2	0.4348	0.4439	0.4393
SGPT-125M	0.3889	0.4079	0.3982
allenai-specter	0.3818	0.4003	0.3908
komninos	0.3263	0.3617	0.3431

Table 6: Performance of models on GPTP dataset.

Model Name	TACC	AUC	AVG
LaBSE	0.9730	0.9586	0.9657
ember-v1	0.9597	0.9527	0.9562
Wartortle	0.9591	0.9521	0.9556
paraphrase-MiniLM-v2	0.9563	0.9455	0.9509
stella-base-en-v2	0.9534	0.9482	0.9508
sentence-t5-large	0.9551	0.9462	0.9506
bge-base-en-v1.5	0.9545	0.9467	0.9506
distiluse-base-v2	0.9522	0.9429	0.9475
e5-base-v2	0.9419	0.9337	0.9378
Venusaur	0.9396	0.9251	0.9323
gte-micro	0.9356	0.9284	0.9320
jina-embeddings-v2-base-en	0.9356	0.9273	0.9314
GIST-Embedding-v0	0.9361	0.9233	0.9297
instructor-base	0.9287	0.9198	0.9242
MedEmbed-small-v0.1	0.9321	0.9155	0.9237
all-MiniLM-L12-v2	0.9315	0.9155	0.9234
Ivysaur	0.9252	0.9137	0.9194
FAB-Ramy-v1	0.9298	0.9084	0.9190
gte-base-en-v1.5	0.9114	0.9010	0.9062
all-mpnet-base-v2	0.9131	0.8963	0.9046
komninos	0.9143	0.8680	0.8905
nomic-embed-text-v1.5	0.8982	0.8812	0.8896
contriever-base-msmarco	0.8964	0.8797	0.8880
snowflake	0.8930	0.8775	0.8852
msmarco-bert-co-condensor	0.8774	0.8674	0.8724
cde-small-v1	0.8843	0.8196	0.8507
gtr-t5-large	0.8619	0.8386	0.8501
allenai-specter	0.8228	0.8060	0.8143
SGPT-125M	0.7664	0.7326	0.7491

Table 7: Performance of models on our PAWS dataset.

Model Name	TACC	AUC	AVG	RANK	GROUP
sentence-t5-large	0.7260	0.6945	0.7099	1	1
FAB-Ramy-v1	0.6939	0.6638	0.6785	6	1
all-mpnet-base-v2	0.6691	0.6463	0.6575	7	1
ember-v1	0.6635	0.6505	0.6569	2	1
nomic-embed-text-v1.5	0.6454	0.6239	0.6345	9	1
e5-base-v2	0.6449	0.6198	0.6321	5	1
GIST-Embedding-v0	0.6375	0.6238	0.6306	3	1
gte-base-en-v1.5	0.6218	0.6144	0.6181	4	1
bge-base-en-v1.5	0.6156	0.6007	0.6081	8	1
instructor-base	0.6009	0.5853	0.5930	10	1
gtr-t5-large	0.5975	0.5795	0.5884	14	1
stella-base-en-v2	0.5862	0.5769	0.5815	15	1
cde-small-v1	0.5795	0.5622	0.5707	16	2
msmarco-bert-co-condensor	0.5609	0.5644	0.5626	19	2
MedEmbed-small-v0.1	0.5490	0.5488	0.5489	12	1
jina-embeddings-v2-base-en	0.5445	0.5500	0.5472	13	1
snowflake	0.5479	0.5416	0.5447	23	2
paraphrase-MiniLM-L12-v2	0.5457	0.5390	0.5423	11	1
contriever-base-msmarco	0.5299	0.5346	0.5322	22	2
LaBSE	0.5389	0.5211	0.5299	20	2
all-MiniLM-L12-v2	0.5287	0.5310	0.5298	18	2
gte-micro	0.4543	0.4702	0.4621	21	2
Ivysaur	0.4476	0.4674	0.4573	24	2
allenai-specter	0.4510	0.4532	0.4521	28	2
Wartortle	0.4391	0.4411	0.4401	17	2
distiluse-base-v2	0.4397	0.4377	0.4387	26	2
SGPT-125M	0.4036	0.4219	0.4125	27	2
komninos	0.3596	0.4050	0.3810	29	2
Venusaur	0.3579	0.3914	0.3739	25	2

Table 8: Performance of models on GPTP dataset, when data for examples that are in between 50% and 80% neutral. RANK: The rank of the model when GPTP has been filtered by our main criterion (contradiction), compare with Table 6. A more coarse view is the GROUP, it shows the binary (better, worse) group that a model is assigned to in the main data (again, c.f., Table 6).

	RANK	MTEB rank (relative)
sentence-t5-large	1	16
ember-v1	2	5
bge-base-en-v1.5	3	4
e5-base-v2	4	9
GIST-Embedding-v0	5	3
FAB-Ramy-v1	6	28
gte-base-en-v1.5	7	2
all-mpnet-base-v2	8	15
instructor-base	9	12
paraphrase-MiniLM-L12-v2	10	19
nomic-embed-text-v1.5	11	7
jina-embeddings-v2-base-en	12	10
MedEmbed-small-v0.1	13	8
stella-base-en-v2	14	6
Wartortle	15	22
LaBSE	16	24
all-MiniLM-L12-v2	17	17
gtr-t5-large	18	13
cde-small-v1	19	1
gte-micro	20	27
msmarco-bert-co-condensor	21	20
contriever-base-msmarco	22	18
snowflake	23	11
Ivysaur	24	14
Venusaur	25	23
distiluse-base-v2	26	29
allenai-specter	27	26
SGPT-125M	28	21
komninos	29	25

Table 9: Comparing our obtained main ranking (Table 3) against the relative ranking on MTEB.