ALIGN-SIM: A Task-Free Test Bed for Evaluating and Interpreting Sentence Embeddings through Semantic Similarity Alignment

Yash Mahajan[1] **Naman Bansal**[1] **Eduardo Blanco**[2] **Santu Karmaker**[3]
[1]Auburn University [2]University of Arizona [3]University of Central Florida
{yzm0034,nzb0040}@auburn.edu, eduardoblanco@arizona.edu, santu@ucf.edu

Abstract

Sentence embeddings play a pivotal role in a wide range of NLP tasks, yet evaluating and interpreting these real-valued vectors remains an open challenge to date, especially in a task-free setting. To address this challenge, we introduce a novel task-free test bed for evaluating and interpreting sentence embeddings. Our test bed consists of five semantic similarity alignment criteria, namely, *semantic distinction, synonym replacement, antonym replacement, paraphrasing without negation, and sentence jumbling*. Using these criteria, we examined five classical (e.g., Sentence-BERT, Universal Sentence Encoder (USE), etc.) and eight LLM-induced sentence embedding techniques (e.g., LLaMA2, GPT-3, OLMo, etc.) to test whether their semantic similarity spaces align with what a human mind would naturally expect. Our extensive experiments with 13 different sentence encoders revealed that none of the studied embeddings aligned with all the five semantic similarity alignment criteria. Yet, most encoders performed highly on the SentEval dataset, a popular task-specific benchmark. This finding demonstrates a significant limitation of the current practice in sentence embedding evaluation and associated popular benchmarks, a critical issue that needs careful attention and reassessment by the NLP community. Finally, we conclude the paper by highlighting the utility of the proposed alignment-based test bed for analyzing sentence embeddings in a novel way, especially in a task-free setting.

1 Introduction

One of the fundamental tasks in NLP is to computationally map sentences into dense vector representations for subsequent analysis. These dense vectors, known as “sentence embeddings", have proven valuable across a wide range of downstream tasks, including translation, question answering, and text classification (Hamann et al., 2019; Gupta et al., 2023; Sarkar et al., 2023, 2022), etc. These embeddings encapsulate the meaning of sentences (and their similarity) in a latent semantic space. However, interpreting and evaluating these dense fixed-size vectors remains an open challenge, particularly in a *task-free* setting.

In this work, we introduce a novel task-free test bed called *ALIGN-SIM* to address these challenges and conduct a comprehensive evaluation of popular sentence embeddings using the same. Our test bed is grounded on five semantic similarity alignment criteria that are intuitive to a human mind: 1) Semantic distinction, 2) Synonym replacement, 3) Antonym replacement, 4) Paraphrasing without negation, and 5) Sentence jumbling. Based on these five criteria, *ALIGN-SIM* systematically tests whether the semantic similarity space of an existing sentence embedding technique aligns with what a human mind would naturally expect.

Computationally, given a sentence S and its corresponding embedding S_x, the basic idea of the test-bed is to (a) perturb S according to a particular criterion to create S' (with embedding S'_x), (b) look at how similar S_x and S'_x are, and compare those observations against the human expected behavior. For example (see Table 1), given the original sentence: “*Fewer than a dozen FBI agents were dispatched to secure and analyze evidence.*”, an example of synonym replacement perturbation is: “Fewer than a dozen FBI agents were ***deployed*** to secure and analyze evidence.”. Obviously, these two sentences are very similar, and intuitively, one would expect their embeddings to be very similar as well. On the contrary, an *Antonym Replacement* or *Sentence Jumbling* perturbation usually shifts/distorts the meaning of the original sentence significantly, and therefore, it is natural to expect a somewhat diverse embedding in the case of Antonym Replacement/Jumbling perturbation. These natural expectations set the basis of our five semantic alignment criteria as well as the *ALIGN-SIM* test bed. From a utility standpoint,

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 7393–7428
November 12-16, 2024 ©2024 Association for Computational Linguistics

Original Sentence: "*Fewer than a dozen FBI agents were dispatched to secure and analyze evidence.*"		
Type of Perturbation	**Example Sentence**	**Expected Encoding**
Paraphrasing	A small number of FBI agents, less than twelve, were sent to secure and examine any relevant evidence.	Similar to Original
Synonym Replacement	Fewer than a dozen FBI agents were *deployed* to secure and analyze evidence.	Similar to Original
Antonym Replacement	*More* than a dozen FBI agents were dispatched to secure and analyze evidence.	Diverse from Original
Paraphrase without Negation	Not more than a dozen FBI agents were deployed for the task of securing and analyzing the evidence	Similar to Original
Sentence Jumbling	Fewer than a *analyze* FBI agents were dispatched to secure and *dozen* evidence.	Diverse from Original

Table 1: Example of the five sentence perturbation proposed to evaluate sentence encoders. **Note**: The example in *Paraphrasing without Negation* serves only as an illustration and has not been utilized in our study.

ALIGN-SIM complements existing task-specific benchmarks by providing a novel way of evaluating and interpreting sentence embedding techniques in terms of how their sentence similarity spaces align with human "expectations" of the same.

To demonstrate the utility of *ALIGN-SIM*, we conducted an extensive evaluation of 13 sentence encoders, including 5 classical models (e.g., Sentence-BERT, Universal Sentence Encoder) and 8 Large Language Model (LLM)-induced (e.g., LLaMA2, GPT3, OLMo) sentence embedding techniques using our test bed. Experimental findings revealed that none of the sentence embedding techniques could fulfill all five semantic alignment criteria, although most of them still achieved high performance on the SentEval dataset (Conneau and Kiela, 2018), a popular task-specific benchmark, which is indeed interesting. In summary, our main contributions are below.

1. We introduce a novel semantic similarity alignment test bed called *ALIGN-SIM* for evaluating and interpreting sentence embedding techniques, which consists of five semantic similarity alignment criteria: 1) Semantic Distinction, 2) Synonym Replacement, 3) Antonym Replacement, 4) Paraphrasing without negation, and 5) Sentence Jumbling.
2. We evaluated thirteen different sentence encoders (5 classical and 8 LLMs) using the *ALIGN-SIM* test bed and found that none of the studied embeddings could align with all the five semantic similarity alignment criteria[1].
3. We curated multiple datasets capturing Synonym/ Antonym/ Jumbled sentence pairs for evaluation (refer to Table 1), which will also help future benchmarking efforts.

2 Related Works

A variety of techniques have been proposed during the last decade to generate embedding for a given sentence. Doc2Vec (Le and Mikolov, 2014) is an unsupervised technique that generates embeddings for variable-length pieces of text and creates unique embeddings for each paragraph in a document. Later, others attempted to learn sentence embedding using auto-encoders (Socher et al., 2011; Hill et al., 2016), (Hu et al., 2017). On the other hand, InferSent (Conneau et al., 2017a) used SNLI (Dolan et al., 2004) and Multi-genre NLI labeled data (Williams et al., 2017) and learned the sentence embedding using a Bi-LSTM with max-pooling and a Siamese network.

More recently, transformer-based models like "Universal Sentence Encoder" (USE) (Cer et al., 2018) were proposed. USE was trained on a combination of supervised and unsupervised NLI data, and it has effectively produced sophisticated sentence embeddings. Sentence BERT (SBert) (Reimers and Gurevych, 2019a) was trained on Wikipedia and news-wire articles and later fine-tuned on SNLI and Multi-Genre NLI datasets. Later, SimCSE (Gao et al., 2021) employed contrastive learning to improve sentence embeddings using a contrastive loss objective. These "classical" models have been trained rigorously on a large corpus of data, and many of them used data parallelisms (Wieting and Gimpel, 2017; Artetxe and Schwenk, 2019b; Wieting et al., 2019a,b), natural language inference (NLI) (Conneau et al., 2017b, 2018; Reimers and Gurevych, 2019b), or a combination of both (Subramanian et al., 2018). These classical models are computationally less demanding due to using a relatively small number of parameters.

A paradigm shift in NLP occurred with the emer-

[1] All embedding techniques tested are based on open-source models except for GPT-3

Model	MR	CR	SUBJ	MPQA	SSTb	TREC	MRPC	Avg
SBERT	83.95	88.98	93.77	89.51	90.01	84.80	**76.28**	86.90
SimCSE	79.69	85.01	93.82	86.91	84.18	87.40	71.19	84.02
USE	75.58	81.83	91.87	87.17	85.68	92.20	69.62	83.42
Infersent	81.10	86.30	92.40	90.2	84.60	88.20	76.20	85.57
LASER	56.14	63.89	67.65	72.36	79.85	89.19	75.19	72.04
Bloom	71.69	80.72	92.09	84.48	84.46	88.80	66.84	81.29
GPTNeo	79.91	83.36	93.48	84.62	88.19	92.40	70.78	84.68
LLaMA-2	83.34	87.15	95.80	87.46	91.65	94.00	65.97	86.48
LLaMA-3	85.14	88.93	96.00	87.94	90.66	94.60	70.09	87.62
GPT3-ada	**88.36**	**93.08**	95.31	**91.29**	**93.63**	**96.00**	73.97	**90.23**
Mistral	83.20	88.08	**96.58**	86.56	91.76	**96.00**	61.39	86.08
OLMo	81.15	88.22	95.80	86.79	90.66	95.60	72.41	87.23
OpenELM	83.73	88.00	95.61	88.38	92.04	**96.00**	71.77	87.93

Table 2: Evaluation of existing sentence encoders on SentEval Benchmark. The accuracy scores are generated using the SentEval toolkit on different classification tasks. Here, GPT3 uses *"text-embedding-ada-002"* for sentence embeddings. The scores are generated using 10-fold cross-validation. **BLUE** and **Pruple** indicate best and second-best performer respectively. More details can be found in Appendix A.3

gence of LLMs (Large Language Models) like GPT-3 (Brown et al., 2020), LLaMA (Touvron et al., 2023), OpenELM (Mehta et al., 2024). While these decoder-only models were originally designed for generation and machine translation tasks (Dankers et al., 2022; Artetxe and Schwenk, 2019a; Lewis et al., 2019; Hu et al., 2017), many researchers recently explored LLMs for their potential in producing high-dimensional sentence embeddings generally fetched from the last hidden layers (Haber and Poesio, 2021; Fournier et al., 2020; Haber and Poesio, 2024; Ethayarajh, 2019). This development has sparked an interest in comparing the embedding spaces of classical (mostly encoder-based) and LLMs (mostly decoder-based).

Additionally, there has been a growing interest in the evaluation and interpretability of sentence encoders and language models. One approach focuses on accuracy-based probing, using classifiers to evaluate model representations (Belinkov and Glass, 2019; Anelli et al., 2022; Voita and Titov, 2020; Conklin and Smith, 2024). Another way to understand embeddings is by showing how they combine different meanings. This line of work involves simple modification on embeddings to represent meaningful relationships, like analogies, at the word level (Mikolov et al., 2013; Pennington et al., 2014; Akter et al., 2023) and the at sentence-level (Liu and Neubig, 2022; Yu and Ettinger, 2020; Dankers et al., 2022; Huang et al., 2023a,b)

Our work is different from previous work in multiple ways: 1) we perform a comparative analysis between classical sentence encoder models and the latest large decoder models, aiming to compare and interpret their embeddings in latent semantic space; 2) we propose a novel test bed by introducing five intuitive semantic alignment criteria; 3) we curated multiple new datasets to facilitate rigorous testing of the proposed 5 semantic alignment criteria.

3 Task-Specific Benchmark: SentEval

We started our investigation by assessing the effectiveness of sentence embeddings on downstream tasks; we evaluated them on the popular SentEval Benchmark (Conneau and Kiela, 2018) (details in Appendix A.3). For this benchmark, we compared five popular classical sentence encoders and eight LLMs. The classical encoder-only models include 1) Universal Sentence Encoder (USE) (Cer et al., 2018), 2) Sentence-BERT (SBert) (Reimers and Gurevych, 2019a), 3) InferSent (Conneau et al., 2017a), 4) SimCSE (Gao et al., 2021). The only classical encoder-decoder model is 5) Language-Agnostic-SEntence Representation (LASER) (Artetxe and Schwenk, 2019a). The eight decoder-only LLMs loaded using huggingface include: 1) GPT3-Ada[2] (OpenAI, 2022), 2) Llama-2-7b-hf (Touvron et al., 2023), 3) Meta-Llama-3-8B (AI@Meta, 2024), 4) GPTNeo (EleutherAI, 2023) 5) Bloom (Scao et al., 2022) 6) Mistral-7B-v0.3 (Jiang et al., 2023) 7) OpenELM-3B (Mehta et al., 2024) 8) OLMo-7B (Groeneveld et al., 2024).

Note that the encoder-based models are trained

[2] We used GPT3 with "text-embedding-Ada-002" model.

to generate sentence embeddings, whereas decoder-only models are originally trained for text generation. More details can be found in Appendix A.4.

Results: The accuracy scores of each sentence embedding model can be found in Table 2. Results reveal a strong performance by LLMs, with GPT-3 achieving the highest average accuracy of 90.23% across datasets. However, classical encoders like SBERT remain highly competitive (86.90%), underscoring their efficiency despite using x1000 times fewer parameters and even surpassing LLaMA2 and a few other LLMs. Furthermore, the close proximity of SimCSE, USE, and Infersent highlights the capabilities of classical encoders. Crucially, all models, both classical and LLMs, perform very competitively with only marginal differences in accuracy scores on the benchmark. But how can we interpret these results? In terms of their latent semantic space, how does SBERT compare with LLaMA2? Indeed, while the task-specific benchmarks are very important and indicative of the extrinsic (task-specific) utility of each sentence embedding technique, evaluating and interpreting the intrinsic properties of sentence embeddings still remains an open challenge to date, especially in a task-free setting.

4 ALIGN-SIM Test Bed and Five Criteria

In this section, we introduce our semantic alignment test bed, *ALIGN-SIM*, for evaluating and interpreting sentence encoders in a novel way, especially in a task-free setting. The test bed comprises five semantic alignment criteria as follows.

1. **Criterion-1 (Semantic Distinction or SD)**: This criterion tests whether a sentence encoder can properly distinguish between a highly semantically related sentence vs. a distinct one. Given a sentence pair (S, S_P) with high semantic overlap and a pair of randomly selected sentences (R_1, R_2), this criteria tests whether a sentence encoder yields similar embeddings for the semantically similar pair (S, S_P) and distinct embeddings for the random pairs (R_1, R_2). Consequently, Criterion-1 tests whether the difference in similarity/distance scores is significant, such that for cosine similarity $Sim(S, S_P) - Sim(R_1, R_2) > \epsilon_{C_1,S}$ and for distance measure such as Normalized Euclidean Distance (NED), whether $NED(S, S_P) - NED(R_1, R_2) < \epsilon_{C_1,N}$, where $C_1, S/N$ is criterion-1 margin reflecting human's natural expectation when (N/S) is the distance/similarity metric.
2. **Criterion-2 (Synonym Replacement)**: Criterion 2 assesses how sentence encoders handle minor lexical variations. To test this, we create a modified sentence (S'_P) by replacing a small number of words in the original sentence (S) with their synonyms. We then test whether a sentence encoder produces similar embeddings for S and S'_P, as synonym substitution typically preserves the overall meaning. The natural expectation here is that the cosine similarity scores should be high and NED scores should be minimal, reflecting the intuition that synonym replacements don't significantly alter sentence meaning.
3. **Criterion-3 (Paraphrase Vs Antonym Replacement):** The third criterion evaluates how sentence encoders capture semantic changes due to antonyms. This test includes three sentences: an original sentence (S), its paraphrase (S'_P), and an antonym-replaced version (S'_A) where one word (verb or adjective) from S is substituted with its antonym. Criterion 3 tests whether a sentence encoder yields embeddings in a way where S is more similar to S'_P than to S'_A. Mathematically, we test whether $Sim(S, S'_P) - Sim(S, S'_A) > \epsilon_{C3,S}$, where $\epsilon_{C3,S}$ denotes the human expectation of minimum margin.
4. **Criterion-4 (Paraphrase without Negation):** Given an input sentence S with negation, criterion 4 tests whether a sentence encoder can identify the semantic equivalence between S and its affirmative paraphrase S'. We quantify this by measuring the similarity or NED between the embeddings of S and S'. A high similarity score (or low NED) would indicate alignment with natural human understanding of negation and its paraphrases without negation.
5. **Criterion-5 (Paraphrase Vs. Sentence Jumbling)** : Criterion 5 assesses how sentence encoders handle word order changes. We compare three sentences: an original sentence (S), its paraphrase (S'_P), and a jumbled version (S'_J) created by randomly swapping word pairs in S. Criterion 5 tests whether a sentence encoder produces embeddings in a way such that S is more similar to S'_P than to S'_J. Mathematically,

we test whether $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$, where $\epsilon_{C5,S}$ denotes the human expected minimum margin.

5 Experiments

5.1 Dataset

In this work, we utilized three publicly available paraphrasing datasets. The datasets are **1) QQP** (Quora Questions Pair) dataset (Chen et al., 2018), **2) PAWS-WIKI** (Paraphrase Adversaries from Word Scrambling-Wikipedia) dataset (Zhang et al., 2019) And, **3) MRPC** (Microsoft Research Paraphrasing Corpus) dataset (Dolan and Brockett, 2005). These datasets feature binary labels: label 1 sentences represent positive pairs/paraphrase pairs, which we term high semantic overlap sentences (POS pairs), acknowledging that not all pairs constitute “true” paraphrases. Label 0 pairs were not directly used due to their semantic relatedness, as they were partial paraphrases in many cases. Instead, we randomly shuffled the non-paraphrased pairs (Label 0) and labeled them as ‘random pairs’ (RND). Finally, Criterion 4 was tested on the **Afin** dataset (Hossain and Blanco, 2022), which contains sentences with negations and their paraphrases without negation, representing challenging paraphrase examples (Details in appendix A.2).

5.2 Implementation Details

All 13 models (mentioned in section 3) were evaluated on the five semantic-alignment criteria designed to assess their alignment with natural human expectations. To facilitate a robust comparison between each model pair, we computed similarity/distance metrics such as *Cosine Similarity* and *Normalized Euclidean Distance (NED)*[3]. These metrics offer insight into the semantic space of each model. For criteria 2, and 4 (refer to Section 4), we normalize the scores using equation 1 and equation 2 (refer to appendix A.5.2) for a fair comparison across models. Both cosine similarity and NED scores were adjusted using model-specific factors: α_{model} for cosine similarity and β_{score} for NED (refer to eq 2). These normalizations help account for model-specific baseline similarities and allow for more meaningful comparisons across different encoders.

Models	**QQP**		**WIKI.**		**MPRC**	
	Sim	**NED**	**Sim**	**NED**	**Sim**	**NED**
USE	85.4	67.7	93.6	71.9	85.8	67.9
SBERT	**91.1**	**70.6**	**96.2**	**73.4**	**88.6**	**69.4**
SimCSE	75.0	62.5	85.7	67.8	77.7	63.9
Infer-Sent	62.8	61.3	64.8	64.4	61.3	61.3
LASER	66.6	62.5	67.8	65.2	65.0	62.7
Bloom	50.1	50.1	50.2	50.1	50.1	50.0
GPTNeo	59.5	54.8	66.2	58.3	61.9	56.0
GPT3-Ada	60.5	55.2	63.3	56.7	61.0	55.5
LLaMA-2	66.7	58.5	75.8	63.0	67.9	59.1
LLaMA-3	68.2	59.1	75.0	62.5	70.9	60.5
Mistral	63.6	57.0	72.4	61.3	69.7	60.0
OpenELM	58.4	54.4	61.3	56.0	59.4	54.9
OLMo	69.3	59.7	78.7	64.4	73.3	61.7

Table 3: **Criterion-1**: Avg. % of samples across different values of epsilon which satisfy the criterion-1 i.e. $Sim(S, S_P) - Sim(S, S_{RND}) > \epsilon_{C1,S}$ and $NED(S, S_P) - NED(S, S_{RND}) < \epsilon_{C1,N}$

$$\alpha_{model} = 1 - \frac{1}{n * |D|} \sum_{i=1}^{n=3} \sum_{j=1}^{|D|} sim(\text{RND-Pairs}) \qquad (1)$$

In Equation 1, n represents the number of datasets and D represents the size of each dataset. The inner average accounts for model randomness by calculating the average "Random Pair (RND)" similarity scores from criterion 1. Random pairs (label 0) are shuffled, and their cosine similarities are averaged across datasets, as shown in (Table 6). For each model, the adjusted score is computed by multiplying the average cosine similarity by equation 1[4]. A similar process is followed for NED scores by equation 2. The adjusting factors penalize models with high similarity scores for random pairs, allowing us to better differentiate between models that align well with human expectations against models that are always overly generous.

5.3 Results

Criterion-1 (Semantic Distinction): For criterion 1, we utilized paraphrase pairs and random pairs from each dataset for our analysis (see section 5.1). Each sentence was encoded using the sentence encoders described in section 3. The results of criterion 1 are shown in Tables 3, 6 and 7. Latter two tables are found in the appendix.

[3] NED is reported in appendix

[4] For criteria 3 and 5 (refer to Section 4), we did not normalize cosine-similarity because we are comparing the raw similarity/distance difference with the expected minimum margin ϵ's. Our results are described below.

Models	QQP			WIKI.			MPRC		
	n=1	n=2	n=3	n=1	n=2	n=3	n=1	n=2	n=3
USE	0.812	0.735	0.671	0.863	0.818	0.777	0.861	0.816	0.771
SBERT	**0.842**	**0.757**	**0.689**	**0.909**	**0.865**	**0.827**	**0.901**	**0.851**	**0.803**
SimCSE	0.628	0.587	0.554	0.658	0.634	0.601	0.658	0.633	0.609
Infer-Sent	0.310	0.296	0.287	0.320	0.311	0.303	0.322	0.315	0.309
LASER	0.412	0.395	0.383	0.427	0.420	0.413	0.426	0.418	0.411
Bloom	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005
GPTNeo	0.274	0.267	0.260	0.278	0.273	0.268	0.279	0.275	0.270
GPT3-Ada	0.257	0.249	0.244	0.263	0.260	0.257	0.263	0.260	0.257
LLaMA-2	0.442	0.393	0.347	0.462	0.432	0.397	0.462	0.430	0.388
LLaMA-3	0.468	0.438	0.412	0.482	0.460	0.435	0.480	0.454	0.424
Mistral	0.413	0.396	0.382	0.420	0.408	0.394	0.419	0.404	0.389
OpenELM	0.218	0.211	0.204	0.220	0.216	0.210	0.220	0.214	0.207
OLMo	0.524	0.488	0.459	0.546	0.524	0.500	0.544	0.518	0.491

Table 4: **Criterion 2**: Adjusted Average **Cosine Similarity** between the Original and the Synonym Replaced Sentence pairs. Columns are grouped by dataset and subdivided by the number of word replacements, $n = \{1, 2, 3\}$. The **blue** and **violet** indicate the best and second-best performer.

These results show varying degrees of alignment with the semantic distinction criteria for different sentence embedding models. Table 3 shows the percentage of samples satisfying Criterion-1 on both evaluation metrics: Cosine Similarity (Sim) and NED. Notably, SBERT and USE show the highest alignment across all datasets for both metrics, indicating they are highly effective in distinguishing between semantically similar and random sentences. In contrast, LLM-induced encoders demonstrated lower alignment compared to classical models with an average of $\sim 61\%$ samples satisfying criterion-1 across all three datasets and two metrics, whereas classical models achieved $\sim 72\%$ alignment under the same conditions.

Further, Tables 6 and 7 (refer to appendix) show the absolute difference in similarity/distance metric scores. Table 6 indicates that SBERT and USE effectively distinguish paraphrase pairs from random pairs, unlike LLMs. One potential reason for the misalignment of LLMs may be attributed to their original design principle: these models are primarily trained as decoder-only models specifically designed for text generation, whereas classical models are encoder-only or encoder-decoder (LASER model) models explicitly trained to produce useful sentence embedding.

This disparity becomes more evident when comparing the findings with the SentEval benchmark (refer to Table 2). While SBERT performed reasonably well on SentEval, it excelled in the semantic distinction criterion. Conversely, GPT-3, a leading performer on SentEval, performed poorly on this criterion. This raises questions on whether the current practice of inducing embeddings from the hidden layers of LLMs is indeed a good idea or not.

Criterion-2 (Synonym Replacement): To create the synonyms perturbed sentence, we first randomly chose n ($n = 1, 2, 3$) words that are verbs or adjectives and replace them with the synonyms retrieved from *WordNET* toolkit (Miller, 1995). Note that these sample pairs have high lexical overlap, which is not usually the case in Criterion 1.

Results of Criterion-2 are presented in Table 4, where it is evident that SBERT and USE exhibit the highest similarity scores for synonym-perturbed pairs across all datasets, with SBERT leading, followed by USE and SimCSE. Although the scores decrease as n increases (which is expected), this trend persists. Remarkably, in comparison to the three classic models, SBERT, USE, and SimCSE, LLM-based embeddings performed poorly, with most failing to meet the criterion (all scores < 0.55) expectation. A similar outcome is observed for the NED metric as well (see Table 8).

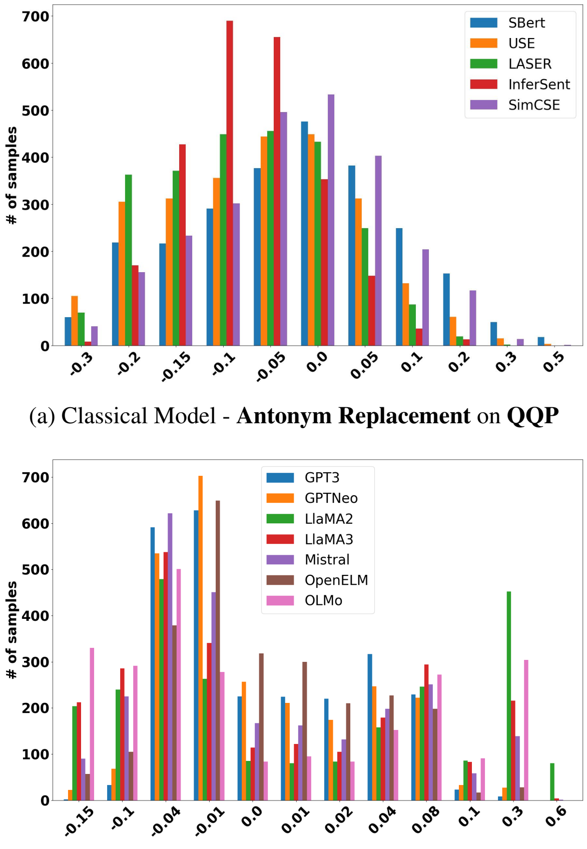
When comparing this criterion and the SentEval benchmark, Table 3 and 4 reveal opposite outcomes. While most LLMs dominated the SentEval benchmark, they yield lower alignment with Criterion 2, which is very interesting.

Criterion-3 (Paraphrase Vs. Antonym Replacement): In the third criterion, we expect that a paraphrase sentence S'_P should be closer to the original sentence S compared to S'_A, which is a perturbed version of S by replacing one word with its antonym from the WordNet (Miller, 1995) toolkit. Figure 1 summarizes the alignment results with

Model	Bloom	GPTNeo	GPT3-Ada	LLaMA2	LLaMA3	Mistral	OpenELM	OLMo
Adjusted Sim. score	0.006	0.267	0.260	0.423	0.441	0.391	0.210	0.491

Model	USE	SBERT	SimCSE	InferSent	LASER
Adjusted Sim. score	0.693	**0.757**	0.58	0.293	0.383

Table 5: **Criterion-4**: Adjusted Avg. Similarity score (equations 1) of negation-affirmative sentence pair sentences from the AFIN dataset. The **blue** and **purple** indicate the best and second-best performer.



(a) Classical Model - **Antonym Replacement** on **QQP**

(b) LLMs - **Antonym Replacement** on **QQP**

Figure 1: **Criterion-3**: The figures demonstrate the **cosine similarity** difference between paraphrase pairs and antonym pairs. The score are calculated based on $Sim(S, S'_P) - Sim(S, S'_A) > \epsilon_{C3,S}$. On the x-axis, the data is grouped into bins, and each bin represents the samples that fall within that $\epsilon_{C3,S}$. This figure represents the QQP dataset. Appendix A.5.3 presents the figures for MRPC, PAWS-WIKI datasets, and NED metric. Note* (We intentionally remove Bloom model from Figure 1 for better visibility and interpretability.)

Criterion-3 by plotting a histogram of the number of sentences that fall into different $\epsilon_{C3,S}$ ranges for cosine, i.e., $Sim(S, S'_P) - Sim(S, S'_A) > \epsilon_{C3,S}$ (NED results provided in appendix).

Upon examining Figure 1a, we notice that all classical sentence encoders display left-skewed histograms (assuming 0 as the center). This skewness suggests their limitations in interpreting the difference between S'_A and S'_P relative to their deviation from the original sentence, S. Such an observation underscores the failure of these encoders to meet Criterion 3, as a majority of the samples lie within the $\epsilon_{C3,S}$ range of -0.3 to 0, with fewer instances exhibiting positive differences.

Turning to Figure 1b a similar trend is evident among LLMs, where they all seem to fail to satisfy the criterion. Observing the distribution of the samples, it is clear that LLMs struggle to properly differentiate between antonym replacements and paraphrases. Bloom (refer to figure 4) performs the worst among the LLMs, with most samples found near zero. This indicates that capturing the nuanced differences in semantically overlapping but opposite pairs remains a significant challenge.

More notably, while LLaMA2 performs poorly compared to other LLMs on SentEval (Table 2), it is the only one that is somewhat able to differentiate between antonym replacements and paraphrases. Similar outcome hold for the other two datasets (refer appendix figures 5,6), and NED metric (refer appendix section A.5.3 figures 7,8, 9). Such observations raise questions about whether the SentEval benchmark is hard enough for testing sentence encoders at a nuanced level.

Criterion-4 (Paraphrase without Negation): To evaluate the paraphrasing without negation criterion, we utilized the Afin dataset (Hossain and Blanco, 2022), which provides negation-affirmation sentence pairs. This dataset allows us to assess how well encoders capture semantic equivalence under lexical alterations involving negation. The expectation for a semantically aligned encoder is to generate high similarity scores (or low NED scores) between these sentence pairs despite the presence of negation in one sentence. This is because the pairs are essentially paraphrases of each other, conveying the same meaning through different lexical structures (refer to Table 1 for examples).

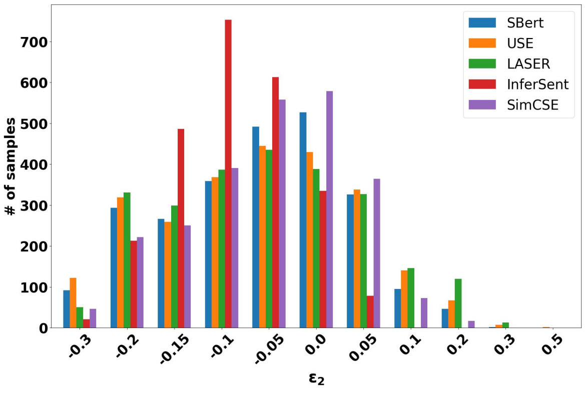
We computed adjusted Cosine Similarity and Normalized Euclidean Distance (NED) scores (refer to Table 5) for all models using these sentence pairs. Note that Criterion 4 is a specialized case of paraphrasing with the added complexity of negation.

As we can see, the SBERT model outperformed all other models by big margins, including LLMs, followed by USE and SimCSE (0.757 vs. 0.693 and 0.58, respectively). As expected, the outcomes are very similar to criteria 1 and 2 for both metrics, as this criterion is essentially a special case of semantically similar sentence pairs. Observing closely reveals that Bloom struggles significantly in this criterion, and the remaining models were sub-optimal. Similar outcomes were observed with the NED metric (refer to Table 9). Surprisingly, classical encoders like LASER and InferSent have comparable performance with other LLMs despite far fewer parameters, smaller size, and architecture. However, it should be noted that LASER and InferSent yield poor alignment with other criteria, and strong alignment with only one criterion does not mean much.

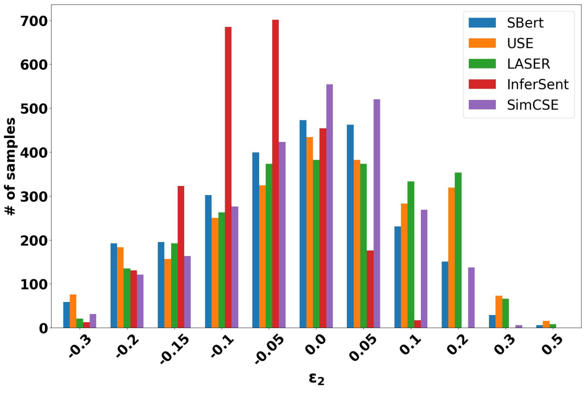
Criterion-5 (Paraphrase Vs. Sentence Jumbling): In this criterion, we expect that when some words are swapped in a sentence, the meaning of the original sentence should be completely destroyed, and the perturbed jumbled sentence S'_J should no longer convey the same meaning as the original sentence S, and thus, it should not be placed close to the original sentence S in the latent semantic space. In contrast, a semantically similar sentence S'_P should be closer to the original sentence S in the same latent space.

To investigate this criterion, an equation similar to Criterion-3 is used for both cosine and NED, except the similarity score of antonym $Sim(S, S'_A)$ is replaced with the similarity of a jumbled sentence, i.e., $(Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S})$. $\epsilon_{C5,S}$ represents the expected minimum margin for this criterion. Next, we plot a histogram of the number of sentences that fall into different $\epsilon_{C5,S}$ ranges. We show the results for the QQP dataset in Figure 2 and 3 (other results are in Appendix A.5.5).

As we notice in Figure 2a, where only one pair of words are swapped ($n = 1$), the classical sentence encoders fail to capture the impact of jumbled words on the sentence similarity task. The majority of the samples lie between the threshold of -0.3 and 0, and only a few samples showed positive differences, suggesting that classical sentence encoders pay little attention to word order. A similar outcome is observed for the other two datasets (MRPC and PAWS) and NED as well (refer to Appendix A.5.5, Figure 19).



(a) Classical Models- **Sentence Jumb.** on **QQP** for $n = 1$



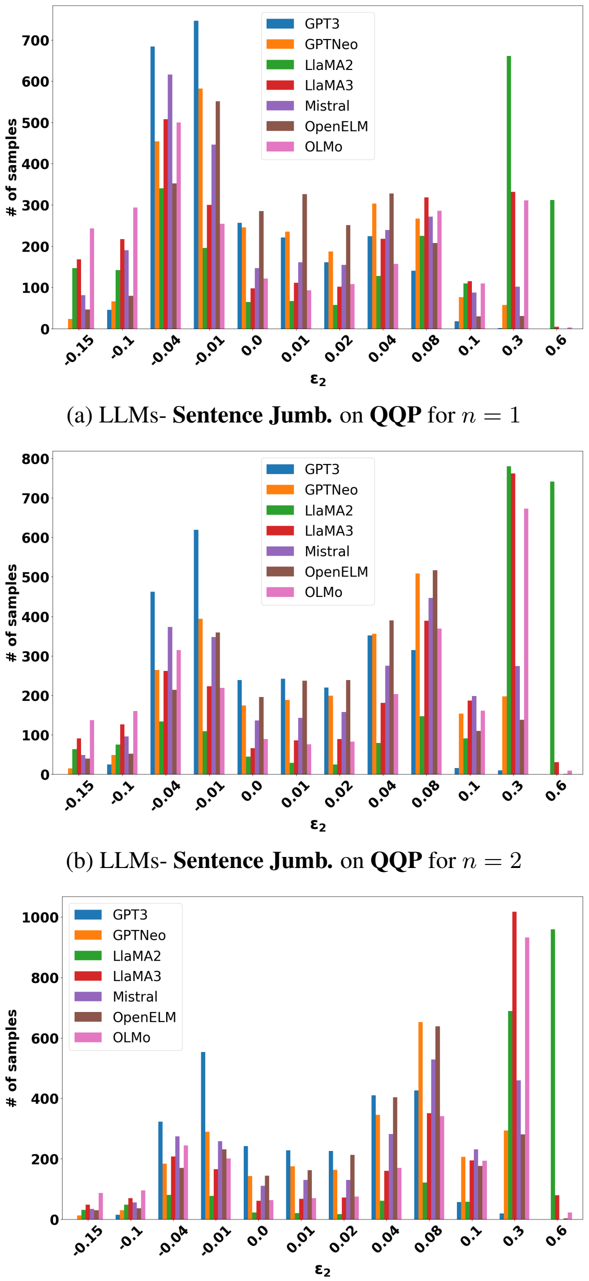
(b) Classical Models- **Sentence Jumb.** on **QQP** for $n = 3$

Figure 2: **Criterion-5**: The figures demonstrate the **cosine similarity** difference for classical models. The score are calculated based on $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$. On the x-axis, the data is grouped into bins, and each bin represents the samples that fall within that $\epsilon_{C5,S}$. Appendix A.5.5 presents the figure for the remaining QQP, MRPC, and PAWS-WIKI dataset. Note: We intentionally remove the Bloom model from the figure for better visibility and interpretability.

In contrast, Figure 3a, which depicts $n = 1$ word pair swapping results for LLMs, reveals that few of these models are notably more adept at capturing the nuances introduced by word order alterations, particularly the LLaMA2,3 and OLMo model. Except for Bloom[5], all LLMs were somewhat able to distinguish between paraphrasing and sentence jumbling. This indicates a marked improvement in the LLMs over the classical models in capturing word order.

Interestingly, LLaMA2's superiority became even more pronounced as n increases (Figure 3b and 3c). A plausible explanation for LLaMA2/3's

[5] Bloom visual represented in Figures 17 and 18



(a) LLMs- **Sentence Jumb.** on **QQP** for $n = 1$

(b) LLMs- **Sentence Jumb.** on **QQP** for $n = 2$

(c) LLMs- **Sentence Jumb.** on **QQP** for $n = 3$

Figure 3: **Criterion-5**: The figures demonstrate the **cosine similarity** difference for LLMs. The score are calculated based on $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$. On the x-axis, the data is grouped into bins, and each bin represents the samples that fall within that $\epsilon_{C5,S}$. Appendix A.5.5 presents the figure for the remaining QQP, MRPC, and PAWS-WIKI dataset. Note: We intentionally remove the Bloom model from the figure for better visibility and interpretability.

and other LLMs' standout performance over classical could be their auto-regressive training process. Indeed, they were trained on a large corpus to predict the next word in the sequence, which may result in better capturing the sensitivity of word orders. A similar outcome was observed with the NED metric (refer to appendix A.5.5).

6 Discussions and Conclusion

In this paper, we introduced a novel semantic alignment test bed, *ALIGN-SIM*, for evaluating and interpreting sentence embeddings in task-free settings. Our framework, grounded on five semantic alignment criteria - *semantic distinction, synonym, and antonym replacement, paraphrasing without negation, and sentence jumbling* - provides a new way of analyzing and comparing the latent semantic spaces of different sentence embedding models.

Our extensive experiments with 13 different sentence encoders, including both classical models and LLM-induced embeddings, revealed a significant contrast between their performance on traditional task-specific benchmarks and their alignment with the proposed five evaluation criteria. Surprisingly, none of the examined sentence embedding models could fulfill all five semantic alignment criteria despite many of them achieving high performance on the SentEval dataset. This discrepancy demonstrates a significant limitation of the current practice in sentence embedding evaluation and associated popular benchmarks, a critical issue that needs careful attention and reassessment by the NLP community. It also raises a concern about which sentence encoder should be used when extensive fine-tuning is unrealistic due to the unavailability of task-specific training data. How should one prioritize between benchmark-specific accuracy and alignment with human expectations when it comes to selecting a particular sentence embedding? More importantly, this work highlights the fact that most works on sentence encoders to date have focused primarily on optimizing accuracy numbers on task-specific benchmarks while ignoring their intrinsic alignment with human's natural expectations.

Nevertheless, this paper demonstrates the utility of our proposed *ALIGN-SIM* test bed as a new paradigm for evaluating sentence embedding models in a task-free setting, complementing task-specific assessments. The test bed, along with the 5 semantic alignment criteria in itself, is a novel contribution, while the curated datasets and extensive experiments that establish an initial task-independent benchmark are also valuable contributions. Our test-bed analysis also reveals that further research is needed to improve sentence encoders that excel in both task-specific benchmarks and alignment with essential linguistic properties expected by humans.

7 Limitation

Our findings are specific to the English language, and our experiments primarily target unsupervised semantic similarity assessments where no training data or prior task-specific knowledge is available. As a result, these findings may not be directly applicable to all downstream NLP tasks. However, in scenarios where training data is unavailable for a particular problem, our results can still serve as a useful guide for selecting an appropriate sentence encoder and designing preliminary experiments.

We focused on three widely used paraphrasing datasets, utilizing only the positive pairs (label 1) for our experiments. However, we observed that a small number of these positive pairs were not true paraphrases, potentially introducing minor noise into our results. Additionally, while the evaluation metrics we employed—Cosine Similarity and NED—offer valuable insights into the semantic alignment of sentence encoders, they may not fully capture all aspects of sentence similarity, such as syntactic structure or deeper semantic meaning.

Finally, our study concentrates on the input-output behavior of sentence encoders and does not explore the internal mechanisms that contribute to their semantic alignment capabilities. Investigating these internal processes could offer further insights into how these models achieve or fail to achieve semantic alignment.

8 Acknowledgements

This work has been partially supported by the National Science Foundation (NSF) Standard Grant Award #2302974 and Air Force Office of Scientific Research Grant/Cooperative Agreement Award #FA9550-23-1-0426. We would also like to thank Auburn University College of Engineering and the Department of CSSE for their continuous support through Student Fellowships and Assistantships.

References

AI@Meta. 2024. Llama 3 model card.

Mousumi Akter, Souvika Sarkar, and Shubhra Kanti Karmaker Santu. 2023. On evaluation of bangla word analogies. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13121–13127. Association for Computational Linguistics.

Vito Walter Anelli, Giovanni Maria Biancofiore, Alessandro De Bellis, Tommaso Di Noia, and Eugenio Di Sciascio. 2022. Interpretability of bert latent space through knowledge graphs. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3806–3810.

Mikel Artetxe and Holger Schwenk. 2019a. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019b. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs. *University of Waterloo*, pages 1–7.

Henry Conklin and Kenny Smith. 2024. Representations as language: An information-theoretic framework for interpretability.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017a. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017b. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Verna Dankers, Christopher Lucas, and Ivan Titov. 2022. Can transformer be too compositional? analysing idiom processing in neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3608–3626, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William Dolan, Chris Quirk, Chris Brockett, and Bill Dolan. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

EleutherAI. 2023. Eleuthera-gtpneo.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*.

Facebook. 2019. Language-agnostic sentence representations (laser).

Louis Fournier, Emmanuel Dupoux, and Ewan Dunbar. 2020. Analogies minus analogy test: measuring regularities in word embeddings. *arXiv preprint arXiv:2010.03446*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.

Vishal Gupta, Ashutosh Dixit, and Shilpa Sethi. 2023. An improved sentence embeddings based information retrieval technique using query reformulation. In *2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, pages 299–304. IEEE.

Janosch Haber and Massimo Poesio. 2021. Patterns of lexical ambiguity in contextualised language models. *arXiv preprint arXiv:2109.13032*.

Janosch Haber and Massimo Poesio. 2024. Polysemy—evidence from linguistics, behavioral science, and contextualized language models. *Computational Linguistics*, 50(1):351–417.

Felix Hamann, Nadja Kurz, and Adrian Ulges. 2019. Hamming sentence embeddings for information retrieval. *arXiv preprint arXiv:1908.05541*.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

Md Mosharaf Hossain and Eduardo Blanco. 2022. Leveraging affirmative interpretations from negation improves natural language understanding. *arXiv preprint arXiv:2210.14486*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168–177, New York, NY, USA. Association for Computing Machinery.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International conference on machine learning*, pages 1587–1596. PMLR.

James Y. Huang, Wenlin Yao, Kaiqiang Song, Hongming Zhang, Muhao Chen, and Dong Yu. 2023a. Bridging continuous and discrete spaces: Interpretable sentence representation learning via compositional operations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14584–14595, Singapore. Association for Computational Linguistics.

James Y. Huang, Wenlin Yao, Kaiqiang Song, Hongming Zhang, Muhao Chen, and Dong Yu. 2023b. Bridging continuous and discrete spaces: Interpretable sentence representation learning via compositional operations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14584–14595, Singapore. Association for Computational Linguistics.

Huggingface. 2023. Huggingface-bigscience-bloom.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Xin Li and Dan Roth. 2002a. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, page 1–7, USA. Association for Computational Linguistics.

Xin Li and Dan Roth. 2002b. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, page 1–7, USA. Association for Computational Linguistics.

Emmy Liu and Graham Neubig. 2022. Are representations built from the ground up? an empirical examination of local composition in language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9053–9073.

Sachin Mehta, Mohammad Hossein Sekhavat, Qingqing Cao, Maxwell Horton, Yanzi Jin, Chenfan Sun, Iman Mirzadeh, Mahyar Najibi, Dmitry Belenko, Peter Zatloukal, et al. 2024. Openelm: An efficient language model family with open-source training and inference framework. *arXiv preprint arXiv:2404.14619*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

OpenAI. 2022. Gpt3-text embedding.

OpenAI. 2023. Openai embeddings.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 271–es, USA. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, page 115–124, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019a. 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.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Souvika Sarkar, Dongji Feng, and Shubhra Kanti Karmaker Santu. 2022. Exploring universal sentence encoders for zero-shot text classification. 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, AACL/IJCNLP 2022 - Volume 2: Short Papers, Online only, November 20-23, 2022*, pages 135–147. Association for Computational Linguistics.

Souvika Sarkar, Dongji Feng, and Shubhra Kanti Karmaker Santu. 2023. Zero-shot multi-label topic inference with sentence encoders and llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 16218–16233. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Richard Socher, Eric Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *Advances in neural information processing systems*, 24.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

NLP Stanford. 2014. Glove embedding.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*.

TensorFlow-Hub. 2018. Use tensorflow-hub.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39:165–210.

John Wieting and Kevin Gimpel. 2017. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732*.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019a. Simple and effective paraphrastic similarity from parallel translations. *arXiv preprint arXiv:1909.13872*.

John Wieting, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019b. A bilingual generative transformer for semantic sentence embedding. *arXiv preprint arXiv:1911.03895*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*.

A Appendix

A.1 Hyper-parameter Search

1. **The exact number of training and evaluation runs :** In this work, our aim was to evaluate each model in a zero-shot setting. Hence, we used pre-trained models and performed our experiments.

A.2 Datasets used

1. **Relevant details such as languages, and number of examples and label distributions:** In this work, we experimented with pairs of English language sentences. All three datasets used in these works are of different sizes, so we randomly sampled each dataset to create a balance between all three datasets. The MRPC dataset consists of a total of 3668 sentence pairs, out of which 1194 pairs of sentences have been labeled 0 and the other 2474 pairs of sentences have been labeled 1. So, to create the balanced dataset, we randomly sampled 1194 pairs of sentences from the dataset having label 1. Subsequently, the QQP and Paws-WIki dataset has 404290 and 49401 pairs of paraphrased and non-paraphrased sentences. So, we randomly sampled nearly 1.2K pair of sentences for each label from each dataset i.e. (2.4 pair of sentences collected from each dataset) for paraphrasing hypothesis testing. For other hypothesis testing (Synonym and Antonym replacement, Sentence Jumbling), we sampled 3.5K sentences (only single sentences, no sentence pairs) from all three datasets. Next, we create the perturbed sentences using the WordNet toolkit for further experiments.

   (a) **QQP**: This publicly available dataset is a collection of question pairs from Quora (Chen et al., 2018) with labels 1 and 0 annotated by humans. Label 1 is assigned when questions in a pair essentially have the same meaning (i.e. paraphrases), and otherwise 0 (i.e. non-paraphrases). In this work, we randomly choose 2.5K samples with label 1 and another randomly shuffled 2.5k samples from label 0 (in total 5k samples).

   (b) **PAWS-WIKI**: This publicly available dataset is a collection of sentence pairs from Wikipedia with high lexical overlapping (Zhang et al., 2019). In this work, we randomly sampled 5K pairs of sentences out of which 2.5K pairs have label 1 and the rest randomly shuffled samples from label 0.

   (c) **MRPC**: This data-set is a collection of sentence pairs collected from newswire articles (Dolan and Brockett, 2005). In total, there are 3.5K sentence pairs out of which 1.1K random pairs from label 0 and the rest are labeled 1 by humans. In this work, we utilize the complete dataset and performed the experiments.

(d) **Afin**: The Affirmative Interpretation of Negation is a dataset comprising approximately 150K sentence pairs, as curated by (Hossain and Blanco, 2022). This dataset features pairs where one sentence contains negation and the other offers an affirmative interpretation of the negated statement (i.e. a paraphrase without negation). It serves as a valuable resource for assessing encoder models' ability to estimate the similarity between such sentences.

A.3 SentEval Toolkit

SentEval (Conneau and Kiela, 2018) is a widely used framework for evaluating the efficacy of sentence embeddings. Here, sentence embeddings are used to perform various classification tasks. Specifically, the SentEval toolkit uses a logistic regression classifier for the datasets we evaluated which deploys a 10-fold cross-validation methodology across a range of classification tasks. The testing fold is then utilized to compute the prediction accuracy of the classifiers.

In this work, we assess the effectiveness of ten distinct sentence encoders on seven datasets from the SentEval benchmark to identify the best one.

1. **MR**: Movie review dataset for binary sentiment classification (Pang and Lee, 2005).
2. **CR**: Sentiment prediction on Product review dataset with binary labels (Hu and Liu, 2004).
3. **MPQA**: An opinion polarity dataset with binary labels (Wiebe et al., 2005).
4. **SSTb**: Stanford Sentiment Treebank dataset with binary labels (Socher et al., 2013).
5. **SUBJ**: Subjective prediction from movie reviews and plot summaries (Pang and Lee, 2004).
6. **TREC**: Fine-grained question-type classification task from TREC (Li and Roth, 2002a).
7. **MRPC**: Mircosoft Paraphrase Corpus from parallel news sources (Li and Roth, 2002b).

A.4 Models Setting

All evaluation criteria and models were loaded and run on NVIDIA Quadro RTX 5000 and NVIDIA RTX A4500 GPUs. The reported results and values were produced on a Linux server. We evaluated various types of encoder-decoder models. Following standard practice in the literature, both encoder and decoder models generate sentence embeddings from the final transformer layers, which consist of contextualized embeddings for each token in the sentence. For both types of models, we averaged the embeddings from the final layer to obtain the sentence embedding. This sentence embedding was used for all our inference tasks. We aim to evaluate open-source models, excluding GPT-3.

1. **USE** (Cer et al., 2018): Universal Sentence Encoder (USE) is a transformer-based encoder only model that encodes the text to a high fixed 512-dimensional fixed-sized vector. The TF2.0 Saved Model (v4) was loaded from (TensorFlow-Hub, 2018) (thumb). The model has been trained to classify: text classification, sentence similarity, and clustering. To encode the sentence we simply use standard TensorFlow USE module.

2. **SBERT** (Reimers and Gurevych, 2019a): Sentence-BERT is a BERT(Devlin et al., 2019) based encoder only model which produces semantically meaningful sentence embeddings. In this work, we used *Sentence-Transformer* library to load the pre-trained model and used *(all-MiniLM-L6-v2)* pre-trained SBERT model for evaluating all criteria. The model has been trained on Wikipedia and Book corpus data to align similar pair sentences, and further fine-tuned on the NLI dataset.

3. **SimCSE** (Gao et al., 2021):SimCSE, which stands for Simple Contrastive Learning of Sentence Embedding, is a encoder only model designed to improve sentence embeddings through a contrastive learning framework. By leveraging contrastive loss, SimCSE aims to enhance the semantic representation of sentences, facilitating better performance in various natural language processing tasks.

4. **InferSent** (Conneau et al., 2017a): The encoder based model produces sentence embeddings having semantic representations of English sentences. In this work, our model used pre-trained GloVe word embeddings (Stanford, 2014) with 840B tokens, 2.2M vocabulary, 300-dimensional vector, and, InferSent version 1 encoder. We have also set the batch size to 64, word embedding dimension size to 300d, and LSTM encoder size to 2048 with max-pooling layers enabled. Additionally, the

model has been trained on the NLI dataset to classify into three categories: entailment, contradiction, and neutral.

5. **LASER** (Facebook, 2019): Language-Agnostic-SEntence Representation (LASER) is a encoder-decoder model built to perform multilingual sentence embedding tasks and trained in 93 different languages. The model used five BI-LSTM layers in the encoder with max-pooling on the last layer to produce embeddings of a sentence. In this study, we used a pre-trained LASER model with its default settings to produce a sentence embedding for a given English sentence.

6. **Bloom**: BLOOM (Scao et al., 2022) is an decoder based autoregressive Large Language Model (LLM) designed to extend text from a given prompt. It has been trained on extensive textual data utilizing substantial computational resources typical of industrial-scale operations. Bloom has 176B parameter. To fetch the embedding from the model we use Huggingface framework (Huggingface, 2023) and fetch the last hidden layer of the model as sentence representation which later gets averaged and then finally served as sentence embedding.

7. **GPTNeo**: The GPTNeo model was released in the EleutherAI/gpt-neo (EleutherAI, 2023). It is a GPT2 like causal language model trained on the Pile dataset. To get a sentence representation, we used hugging face framework (Huggingface, 2023). After loading a model, we utilized last hidden layer to fetch sentence representation. Next, we averaged the embedding which server as sentence embedding.
8. **GPT3**-Ada: We used GPT3 (OpenAI, 2022) text-embedding-ada-002 model which is trained for text search, text similarity, and code search. We generate embedding using OpenAI API (OpenAI, 2023). The output dimension produced by the model is 1536.

9. **LLaMA2**: The LLaMA2 (Touvron et al., 2023) model is a collection of pre-trained and fine-tuned large language models (LLMs) ranging in scale from 7 billion to 70 billion parameters. In this work, we used the 7B parameter for encoding text. We utilized the HuggigFace framework with LLaMA2 weights and generated the encodings. To generate an embedding vector, the decoder processes input tokens (embeddings) to generate corresponding output embeddings, and we computed the mean of these output embeddings to serve as the sentence embedding, following standard practice. The final output dimension vector was 4096.

10. **LLaMA-3**: The LLaMA-3 (AI@Meta, 2024) is a decoder model, which is a pre-trained and instructed fine-tuned language model released in 8B and 70B sizes. In this work, we used the 8B pre-trained model, and the HuggingFace framework was utilized to load the model. We performed inference testing on our proposed criteria using this model. To encode a sentence, we used a similar approach as mentioned in the LLaMA-2 model. The model generates a 4096-dimensional embedding vector.

11. **Mistral**: Mistral (Jiang et al., 2023) is an open source model multilingual model available in various sizes. In this work, we used the Mistral-7B-v0.3 model which we loaded using the HuggingFace Framework. The model produced an embedding size of 4096.

12. **OLMo**: Open Language Model by Allen Institute of AI (AI2) is an open-source model. OLMo released different sizes and we use the 7B model. We used the Huggingface framework to load the model. We use the same process as above to generate the embedding. The model produces a 4096 embedding dimension.

13. **OpenELM**: The Open Efficient Language Model was released by Apple in various sizes. In this work, we use the openELM-3b model which is their biggest model. We use a similar setup as other decoder-only models. The output embedding size is 3072 dimensions.

A.5 Results

A.5.1 Criterion-1: Semantic Distinction

This criterion tests whether a sentence encoder can properly distinguish between a highly semantically related sentence vs. a distinct one. In Table 6, pos represent the similarity of paraphrase sentences,

Metric	QQP			WIKI			MRPC		
	Pos	**RND**	**Diff**	**Pos**	**RND**	**Diff**	**Pos**	**RND**	**Diff**
USE	0.831	0.125	0.707	0.956	0.079	0.878	0.788	0.075	0.713
SBERT	0.866	0.042	**0.823**	0.974	0.041	**0.933**	0.835	0.063	**0.772**
SimCSE	0.873	0.373	0.500	0.975	0.264	0.711	0.873	0.318	0.555
Infer-Sent	0.891	0.636	0.255	0.974	0.679	0.295	0.923	0.698	0.225
LASER	0.839	0.505	0.333	0.970	0.616	0.355	0.874	0.574	0.300
Bloom	0.999	0.995	0.004	1.000	0.994	0.005	0.999	0.996	0.003
GPTNeo	0.951	0.762	0.189	0.985	0.665	0.320	0.963	0.727	0.236
GPT3-Ada	0.950	0.739	0.211	0.989	0.724	0.266	0.958	0.739	0.219
LLaMA-2	0.882	0.549	0.333	0.958	0.441	0.517	0.916	0.557	0.359
LLaMA-3	0.890	0.530	0.360	0.965	0.472	0.493	0.912	0.495	0.417
Mistral	0.918	0.651	0.267	0.968	0.524	0.444	0.921	0.528	0.394
OpenELM	0.954	0.790	0.164	0.989	0.767	0.221	0.958	0.772	0.186
OLMo	0.870	0.487	0.383	0.956	0.385	0.571	0.893	0.427	0.466

Table 6: **Criterion-1**-Semantic Distinction: Average **Cosine Similarity** for . Here, Positive (Pos.) is semantically high overlap pairs and RND is random pairs. Diff is the difference of "Pos" and "RND". **Blue** and **Violet**indicate best and second-best performer respectively.

Metric	QQP			WIKI			MRPC		
	Pos	**Neg**	**Diff**	**Pos**	**Neg**	**Diff**	**Pos**	**Neg**	**Diff**
USE	0.0842	0.4377	-0.3534	0.0218	0.4606	-0.4387	0.1059	0.4624	-0.3565
SBERT	0.0672	0.4789	**-0.4117**	0.0130	0.4796	**-0.4666**	0.0827	0.4687	-0.3860
SimCSE	0.0638	0.3141	-0.2503	0.0125	0.368	-0.3555	0.0638	0.341	-0.277
Infer-Sent	0.0919	0.3165	-0.2245	0.0258	0.3137	-0.2879	0.0794	0.3053	-0.2259
LASER	0.1172	0.3672	-0.2500	0.0254	0.3296	-0.3041	0.1086	0.3627	-0.2541
Bloom	0.0008	0.0031	-0.0023	0.0003	0.0035	-0.0032	0.0006	0.0024	-0.0018
GPTNeo	0.0253	0.1213	-0.0960	0.0076	0.1701	-0.1625	0.0190	0.1390	-0.1199
GPT3-Ada	0.0252	0.1307	-0.1055	0.0053	0.1383	-0.1330	0.0212	0.1307	-0.1095
LLaMA-2	0.0602	0.2281	-0.1679	0.0214	0.2822	-0.2608	0.0431	0.2241	-0.1810
LLaMA-3	0.0555	0.2359	-0.1804	0.0181	0.2654	-0.2474	0.0444	0.2532	-0.2088
Mistral	0.0249	0.1791	-0.1363	0.0164	0.2412	-0.2248	0.0407	0.2394	-0.1987
OpenELM	0.0256	0.1129	-0.0874	0.0063	0.1245	-0.1182	0.0223	0.1203	-0.0980
OLMo	0.0658	0.2579	-0.1921	0.0221	0.3090	-0.2869	0.0542	0.2880	-0.2338

Table 7: **Criterion-1**: We expect the difference of NED score between semantically similar pairs and random pairs should be negative. Higher the Negative score better the models is as its a distance measure. The **blue** and **purple** indicate the best and second-best performer.

and RND represents the similarity of random pairs. A good sentence encoder should show a significant difference, with pos being much higher than RND.

From the table, we observe that SBERT and USE dominated all other models, especially LLMs. Among LLMs, OLMo and the LLaMA family performed well but were still far behind SBERT. Evaluating this criterion using the NED metric (refer to Table 7), we found a similar trend: classical models outperformed LLMs. We believe that this is because classical models are mostly trained to align similar sentences, thus performing well on this task, whereas LLMs are trained for text generation and therefore produce more balanced embeddings.

A.5.2 Criterion-2: Synonym Replacement

We expected synonym-perturbed sentences to be more semantically similar to the original sentences. Tables 4 and 8 represent the cosine similarity and NED distance between these pairs, showing a decrease in similarity and an increase in distance as n increases. However, we observed that models like BLOOM and others produced inflated similarity scores before normalization. To address this

Models	QQP			WIKI.			MPRC		
	n=1	n=2	n=3	n=1	n=2	n=3	n=1	n=2	n=3
USE	0.024	0.043	0.043	0.011	0.022	0.022	0.011	0.023	0.023
SBERT	0.027	0.048	0.048	0.011	0.021	0.021	0.013	0.025	0.025
SimCSE	0.014	0.024	0.024	0.006	0.012	0.012	0.006	0.012	0.012
Infer-Sent	0.013	0.024	0.024	0.008	0.015	0.015	0.007	0.013	0.013
LASER	0.013	0.024	0.024	0.005	0.010	0.010	0.006	0.012	0.012
Bloom	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997
GPTNeo	0.859	0.861	0.861	0.858	0.859	0.859	0.857	0.858	0.858
GPT3-Ada	0.869	0.871	0.871	0.868	0.868	0.868	0.868	0.868	0.868
LLaMA-2	0.766	0.779	0.779	0.761	0.769	0.769	0.761	0.769	0.769
LLaMA-3	0.757	0.765	0.765	0.753	0.759	0.759	0.754	0.761	0.761
Mistral	0.785	0.790	0.790	0.783	0.787	0.787	0.783	0.787	0.787
OpenELM	0.883	0.885	0.885	0.882	0.883	0.883	0.882	0.884	0.884
OLMo	0.726	0.735	0.735	0.720	0.726	0.726	0.721	0.727	0.727

Table 8: **Criterion 2:** Adjusted Normalized NED score between the Original and the Synonym Replaced Sentence pairs. Columns are grouped by dataset and subdivided by the number of word replacements, $n = \{1, 2, 3\}$.

issue, we used adjustment factors α and β for cosine similarity and NED metrics, respectively. The purpose of these adjustments is to penalize the inflated scores produced by the models and provide a more accurate interpretation. To normalize the values, we used the "RND" pair similarity score from Criterion-1 (refer to Table 6 and)7 in the formulation of α and β as shown below. We applied α and β to normalize the similarity and distance scores for Criterion-2 and Criterion-4 only.

Adjustment Factor (α) for cosine: To account for the randomness between pairs in each dataset, we first find the similarity score of random pairs, denoted as "RND"(see table 6). For each model, we calculate the similarity scores by shuffling the "RND" pairs and then taking the average similarity score (average randomness score) corresponding to each dataset. Next, we adjusted the average similarity score by multiplying it by (α_{model}).

$$\alpha_{model} = 1 - \frac{1}{n * |D|} \sum_{i=1}^{n=3} \sum_{j=1}^{|D|} sim(\text{RND-Pairs})$$

The average of randomness scores refers to the average of average similarity scores across all three datasets and all models for Neg pairs. This is proven to eliminate the effect of model randomness. Finally, we report the results in Tables(4, 5 and 6).

Adjustment Factor (β) for NED: To account for the randomness between pairs in each dataset, we first find the distance between random pairs, denoted as "RND"(see table 7). For each model, we calculate the distance metric by shuffling the "RND" pairs and then taking the average similarity score (average randomness score) corresponding to each dataset The equation is defined as:

$$\beta_{\text{model}} = 1 - ((1 - \text{NED}_{\text{score}}) \times \left(1 - \frac{1}{n * |D|} \sum_{i=1}^{n} \sum_{j=1}^{|D|} \text{sim} \left({}^{\text{RND-}}_{\text{Pairs}} \right) \right)) \quad (2)$$

Equation 2 components breakdown

1. $(1 - \text{NED}_{\text{score}})$ score: This term measures how far the model is from the maximum distance, i.e., how similar it is rather than how dissimilar.
2. Average Distance calculation: $\frac{1}{n*|D|} \sum_{i=1}^{n=3} \sum_{j=1}^{|D|} sim(\text{RND-Pairs})$, this term calculates the average distance over all datasets and all data points within those datasets. It effectively normalizes the total distance by the product of the number of datasets and the size of each dataset.
3. In inner *(1− above equation expression)* takes the complement of the normalized average distance, then takes the complement again, which simplifies to just the normalized average distance.
4. Combine all, the terms from the NED score with the average distance measure, taking a

Model	USE	SBERT	SimCSE	Infer-sent	LASER	Bloom	GPTNeo	GPT3-Ada	LLaMA2	LLaMA-3	Mistral	OpenELM	OLMo
Adjusted Sim. score	0.053	0.049	**0.026**	0.032	0.034	0.997	0.86	0.87	0.771	0.764	0.791	0.885	0.734

Table 9: **Criterion-4**: Adjusted Normalized Euclidean Distance (using equations 2) of negation-affirmative sentence pair sentences from the AFIN dataset. The **blue** and **violet** indicate the best and second-best performer.

product of these two terms. The outer 1 - term is used to compute the final β scores.

A.5.3 Criterion-3:Paraphrase Vs Antonym Replacement

Figures 5 and 6 elucidate the performance of encoder models on the PAWS-WIKI and MRPC datasets, respectively. A discernible observation is the classical encoders' struggle to differentiate between opposing sentence pairs, underscoring their limitations in handling foundational linguistic tasks. Contrarily, while LLMs also face challenges, the LLaMA2, LLaMA3 and OLMo model evidences a modest edge over its classical counterparts. On the whole, our findings suggest that while LLMs have achieved incremental advancements over classical models in Criterion-3, substantial opportunities for refinement remain.

A.5.4 Criterion-4: Paraphrase without Negation

This case is a special type of paraphrasing data where one sentence contains a negation and the other is its affirmative paraphrase. We expect the model to produce similar embeddings for these pairs. Interestingly, in the NED metric, the SimCSE model surpasses SBERT. This is likely due to SimCSE's training with a contrastive loss function, which aims to separate unlike pairs during training. However, when comparing NED table 9 with cosine table 5, we observe some discrepancies, while in NED, SimCSE leads. This indicates that while the angle between the vectors is closer for SBERT, the magnitude difference is lower for SimCSE. This suggests that SimCSE effectively captures magnitude differences and could be more useful than cosine similarity for tasks involving negation sentences.

A.5.5 Criterion-5: Paraphrase Vs Jumble Sentence

The results of the cosine similarity and NED difference for the Jumble Sentence task are shown in Figure [10 - 18] for cosine and for NED Figure [19 - 21]. All figures showcase the model's ability to capture semantic meaning when the words are swapped by order of *'n'* i.e. $n = 1, 2, 3$ across all three datasets. The difference score is calculated as $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$ and $NED(S, S'_P) - NED(S, S'_J) < \epsilon_{C5,N}$ for NED. All sentence encoders were evaluated on three datasets, and the results suggest that the classic models struggle to capture the word order of sentences whereas LLMs show some progress over classic models. The figures display the number of samples with a difference in cosine similarity score greater than $\epsilon_{C5,S/N}$.

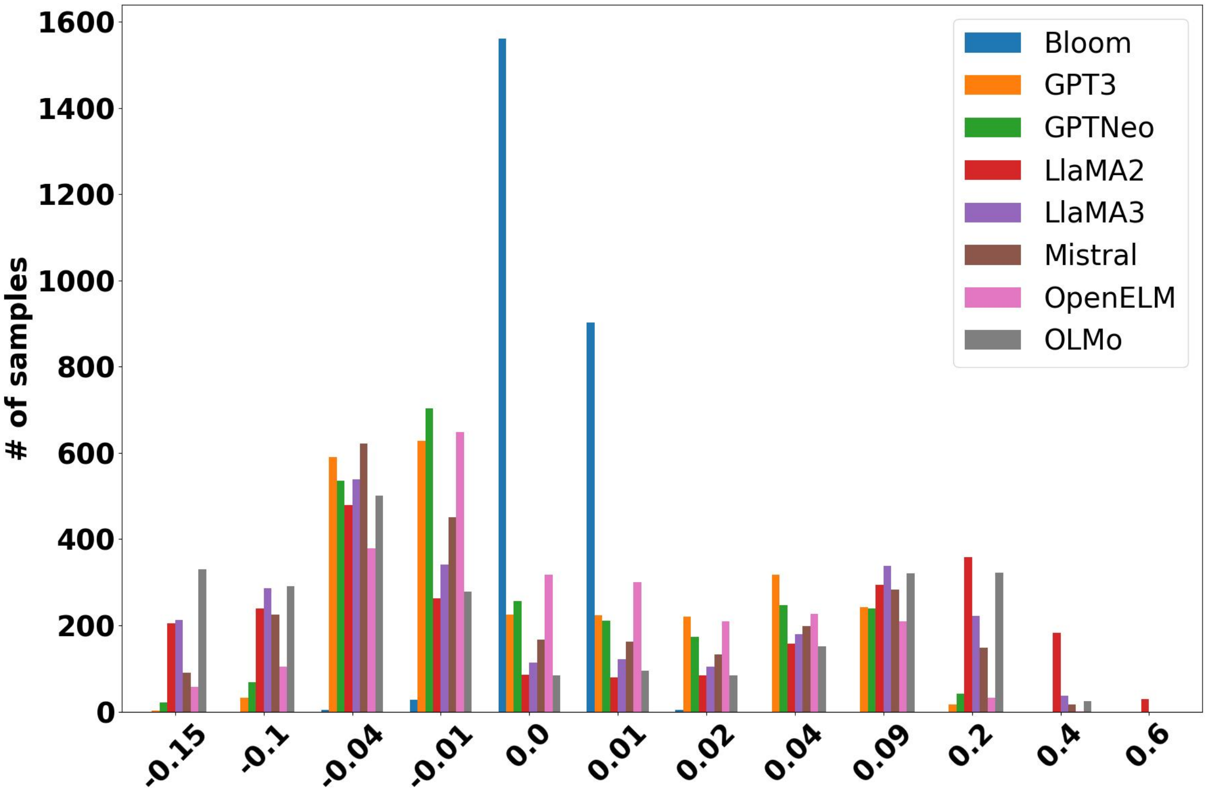
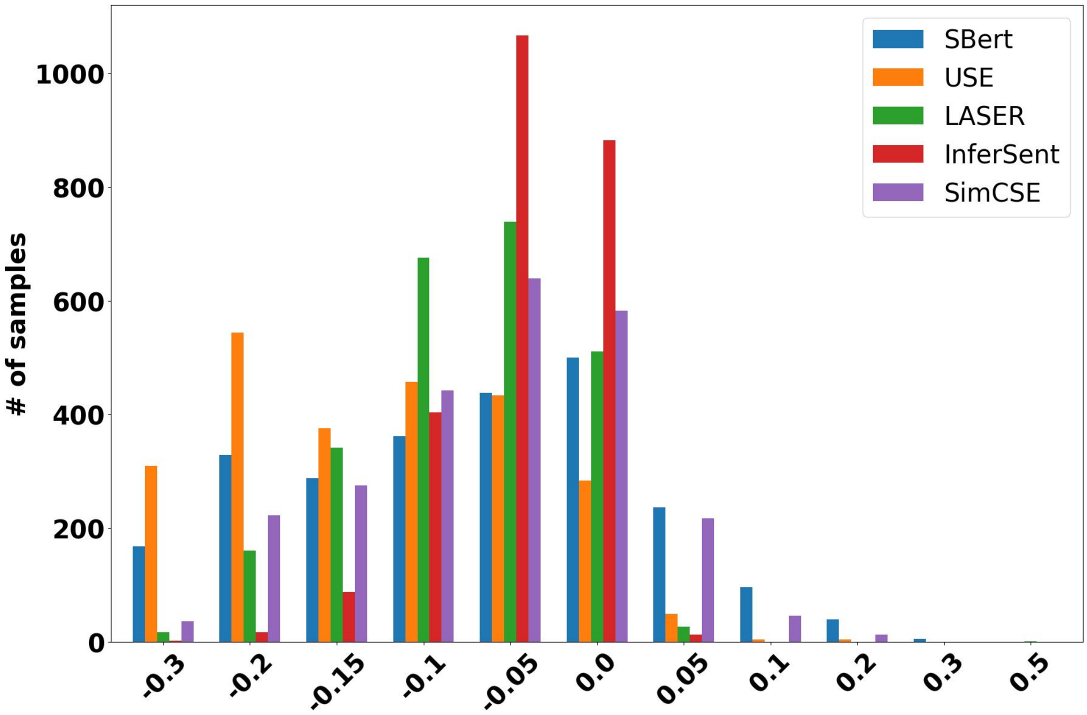
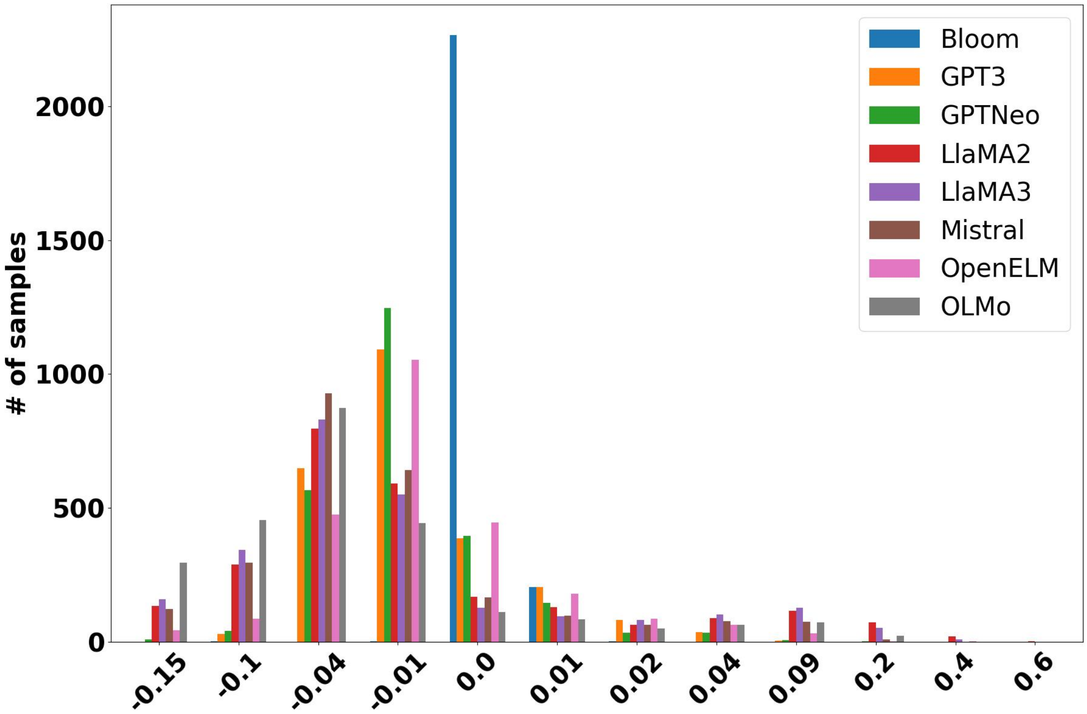


Figure 4: **Criterion-3**: LLM Model- **Antonym Replacement** Task. showcasing the difference in **cosine similarity** score using the formula $Sim(S, S'_P) - Sim(S, S'_A) > \epsilon_{C3,S}$, where $\epsilon_{C3,S}$ denotes the expected minimum margin of differentiation on x-axis. This figure includes Bloom model.

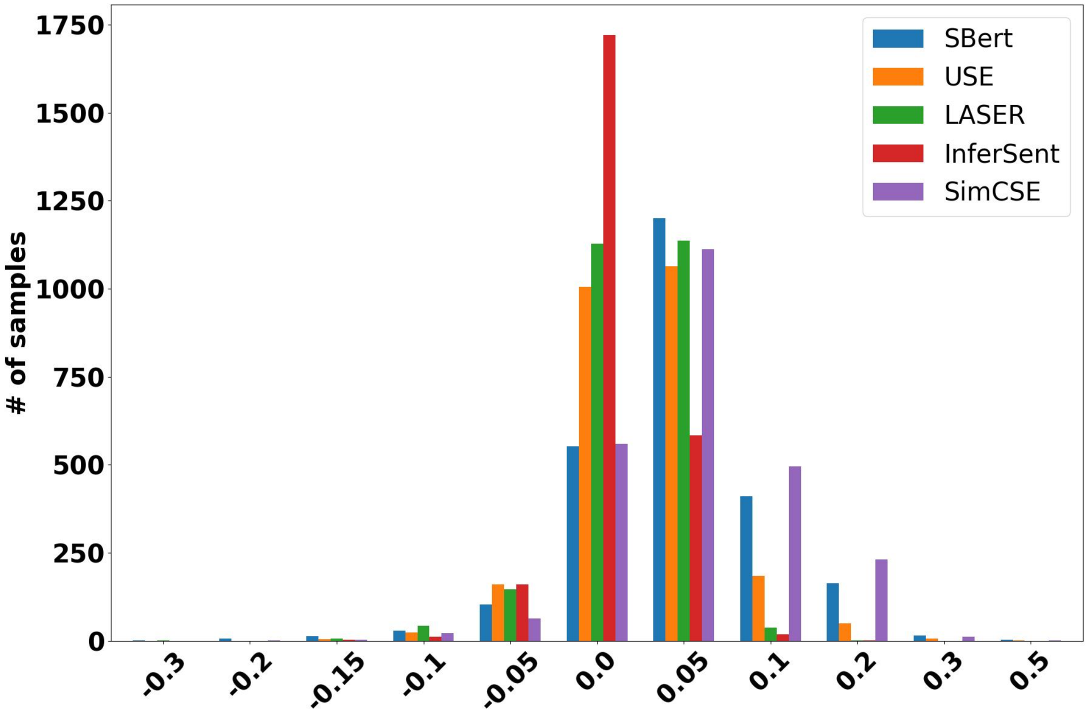


(a) **Classical Model** - **Cosine Metric** on **Antonym** Replacement Task on **MRPC** dataset

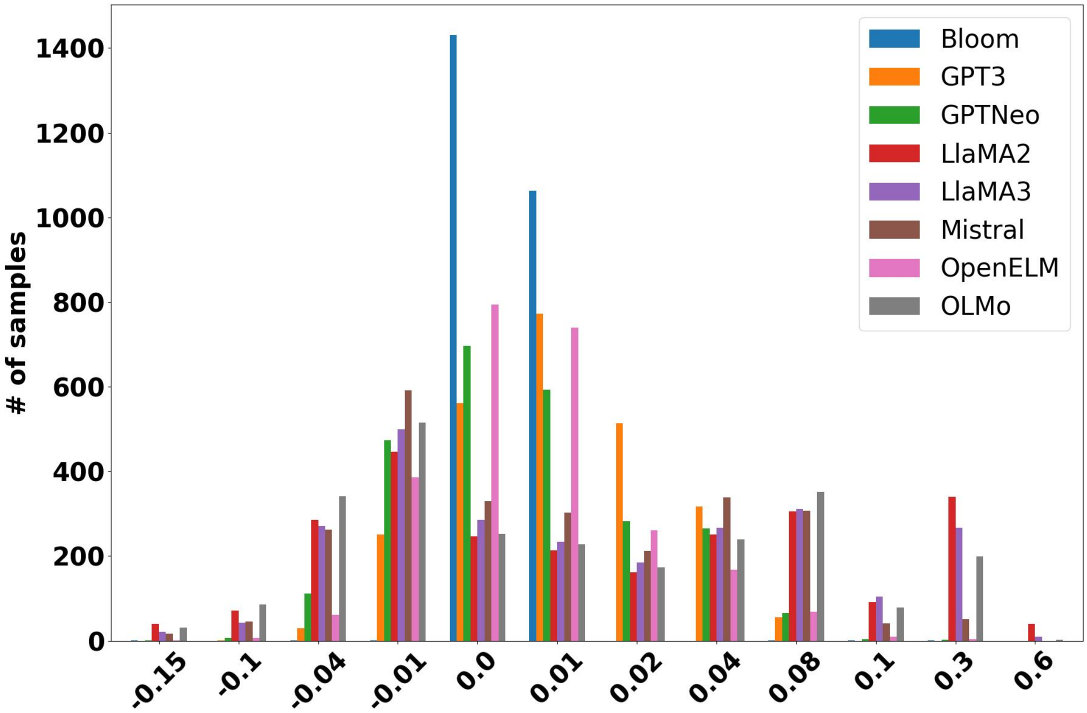


(b) **LLMs** - **Cosine Metric** on **Antonym** Replacement Task on **MRPC** dataset

Figure 5: **Criterion-3**: The figure of MRPC dataset showcasing the difference in cosine similarity score using the formula $Sim(S, S'_P) - Sim(S, S'_A) > \epsilon_{C3,S}$, where $\epsilon_{C3,S}$ denotes the expected minimum margin of differentiation. Figure (a) Classical Encoders and (b) LLMs. It highlights their ability to distinguish between a sentence and its antonym counterpart on MRPC.

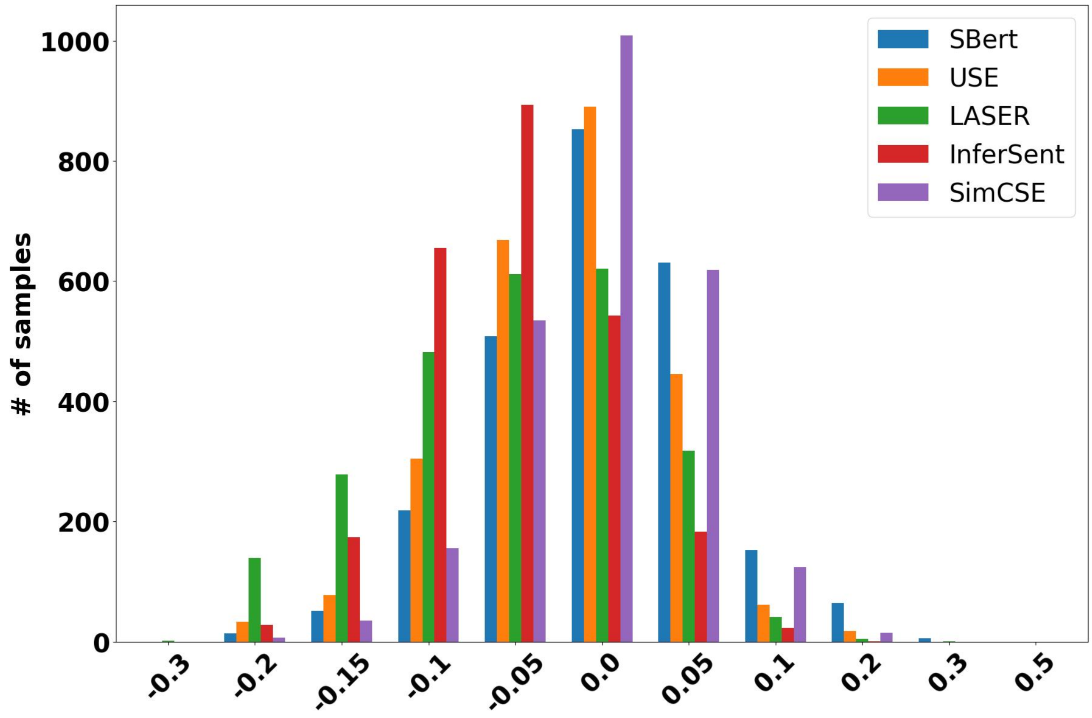


(a) **Classical Model** - **Cosine Metric** on **Antonym** Replacement Task on **PAWS-WIKI** dataset

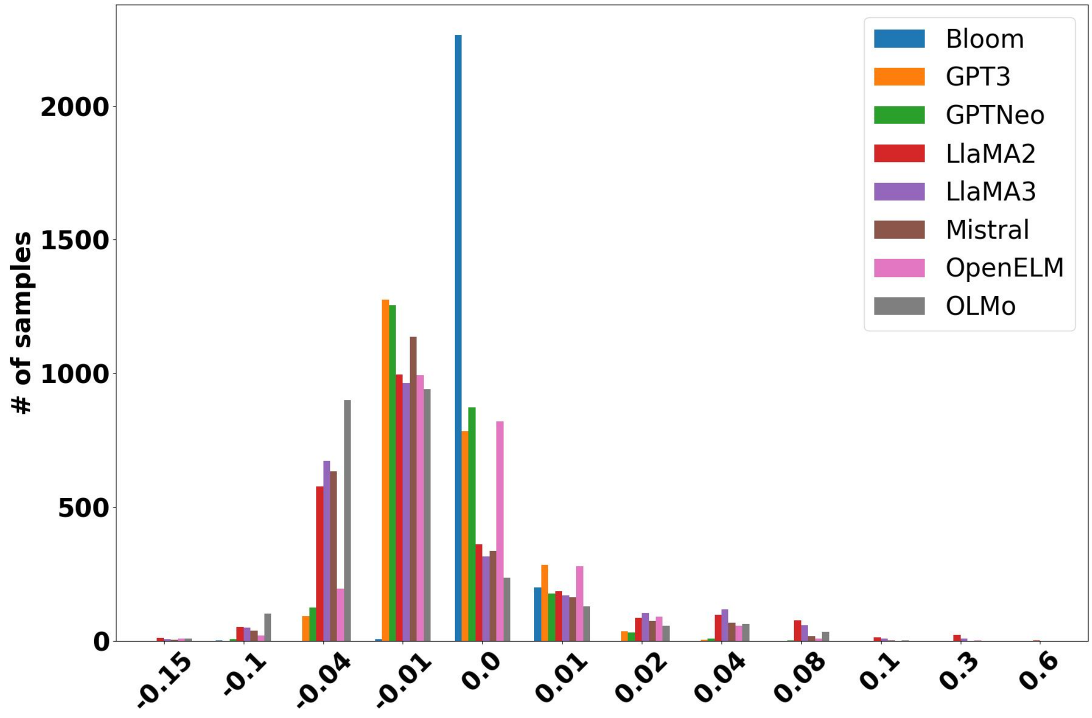


(b) **LLMs** - **Cosine Metric** on **Antonym** Replacement Task on **PAWS-WIKI** dataset

Figure 6: **Criterion-3**: The figure of PAWS-WIKI dataset showcasing represent the difference in cosine similarity score using the formula $Sim(S, S'_P) - Sim(S, S'_A) > \epsilon_{C3,S}$, where $\epsilon_{C3,S}$ denotes the expected minimum margin of differentiation. Figure (a) Classical Encoders and (b) LLMs. It highlights their ability to distinguish between a sentence and its antonym counterpart on PAWS-WIKI.

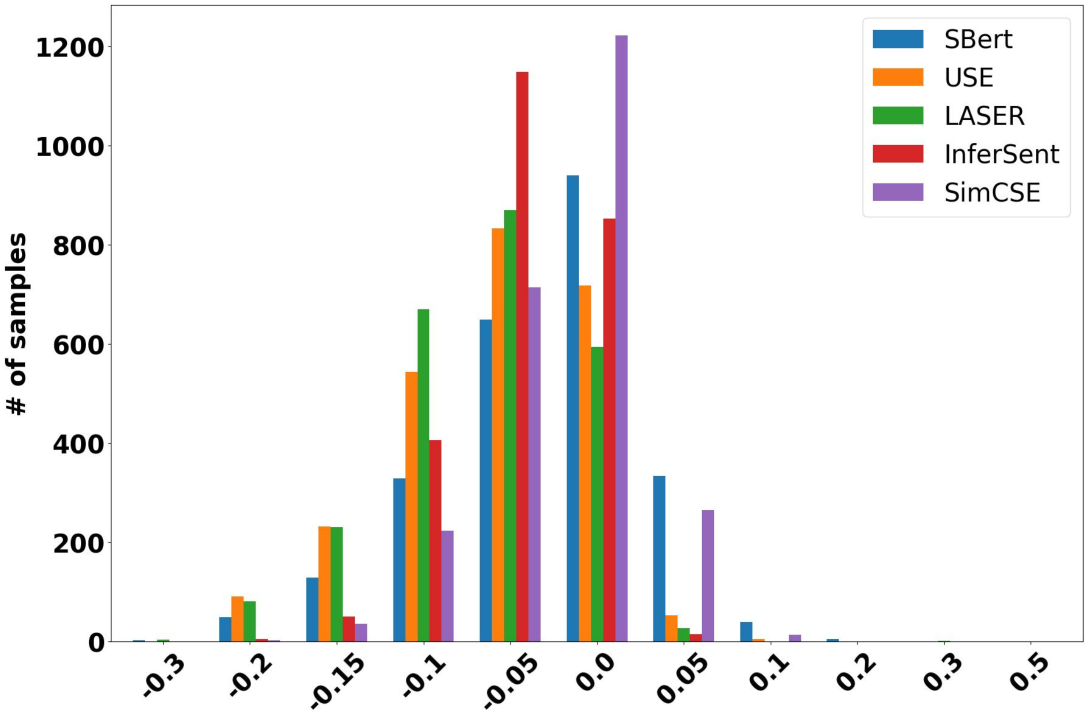


(a) **Classical Model** - **NED Metric** on **Antonym** Replacement Task on **QQP** dataset

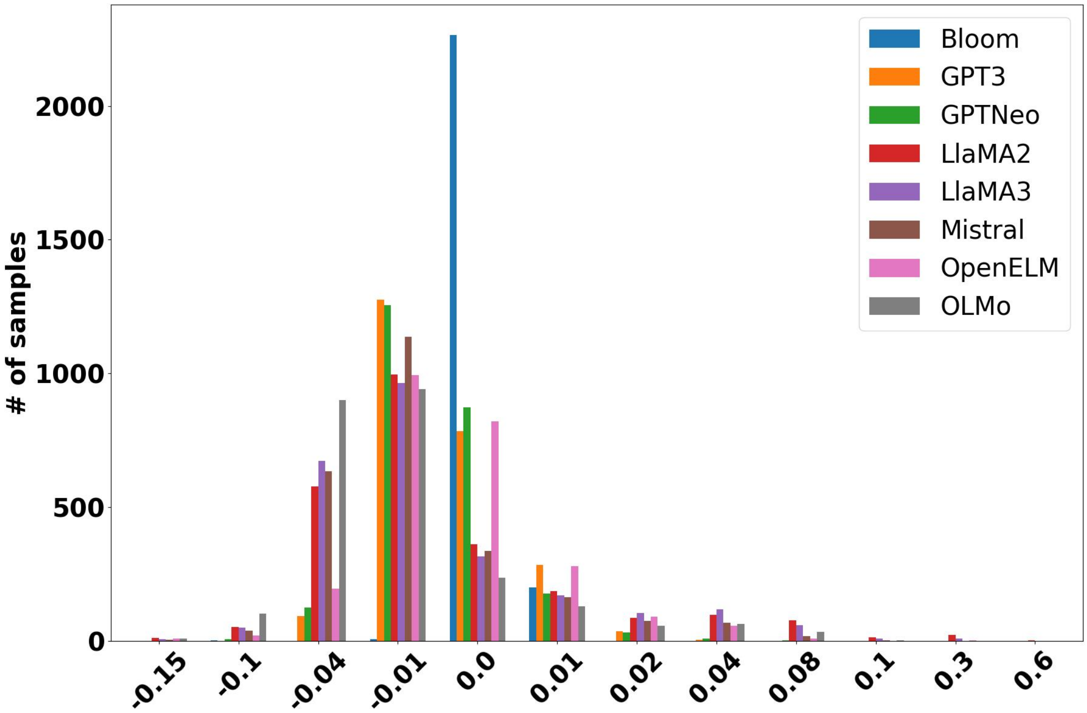


(b) **LLM** - **NED Metric** on **Antonym** Replacement Task on **QQP** dataset

Figure 7: **Criterion-3**: The figure of **QQP** dataset showcasing represent the difference in **NED distance** using the formula $NED(S, S'_A) - NED(S, S'_P) > \epsilon_{C3,N}$, where $\epsilon_{C3,N}$ denotes the expected minimum margin of differentiation. Note, that positive side represent better model.

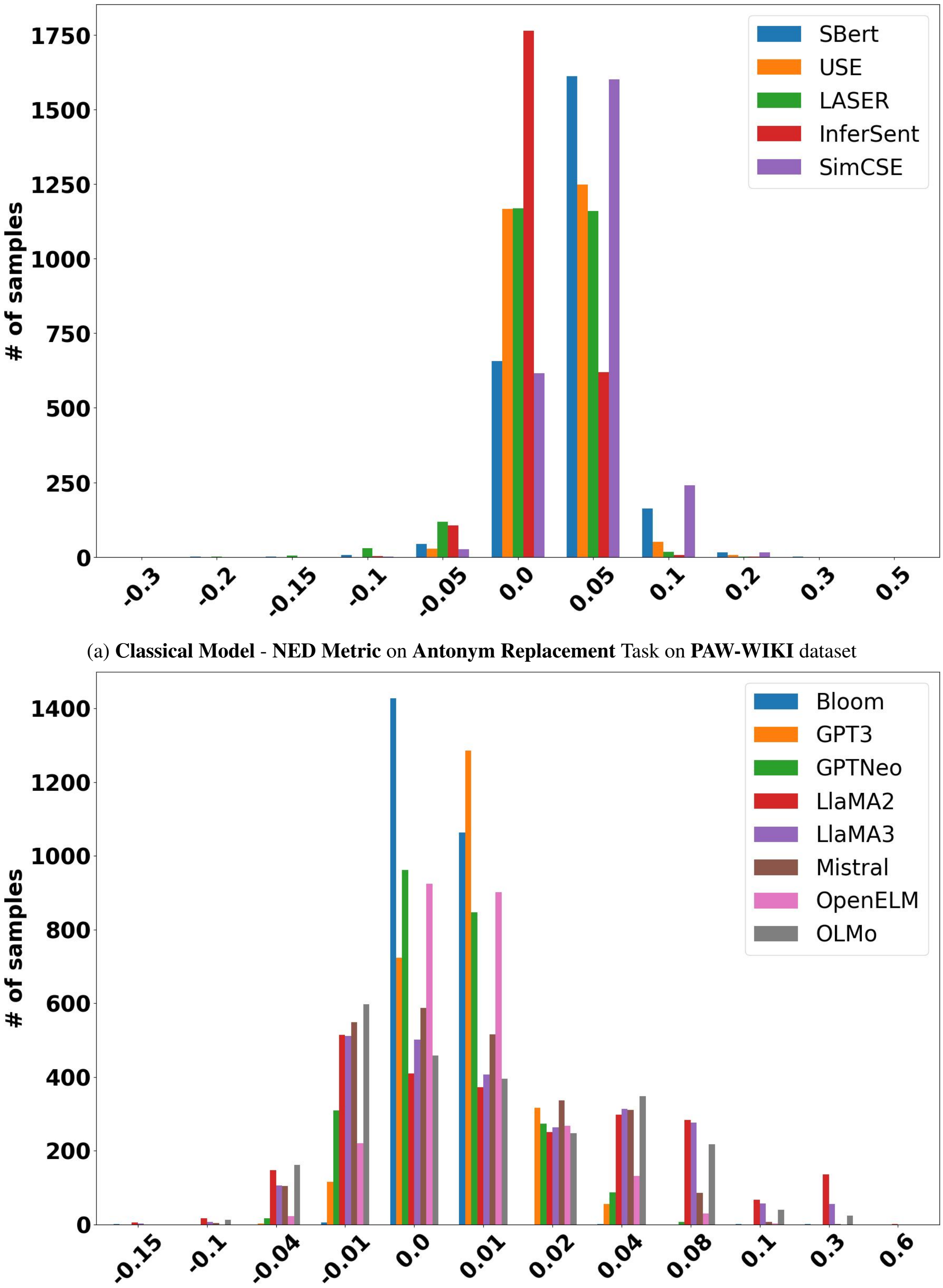


(a) **Classical Model** - **NED Metric** on **Antonym** Replacement Task on **MRPC** dataset



(b) **LLMs** - **NED Metric** on **Antonym** Replacement Task on **MRPC** dataset

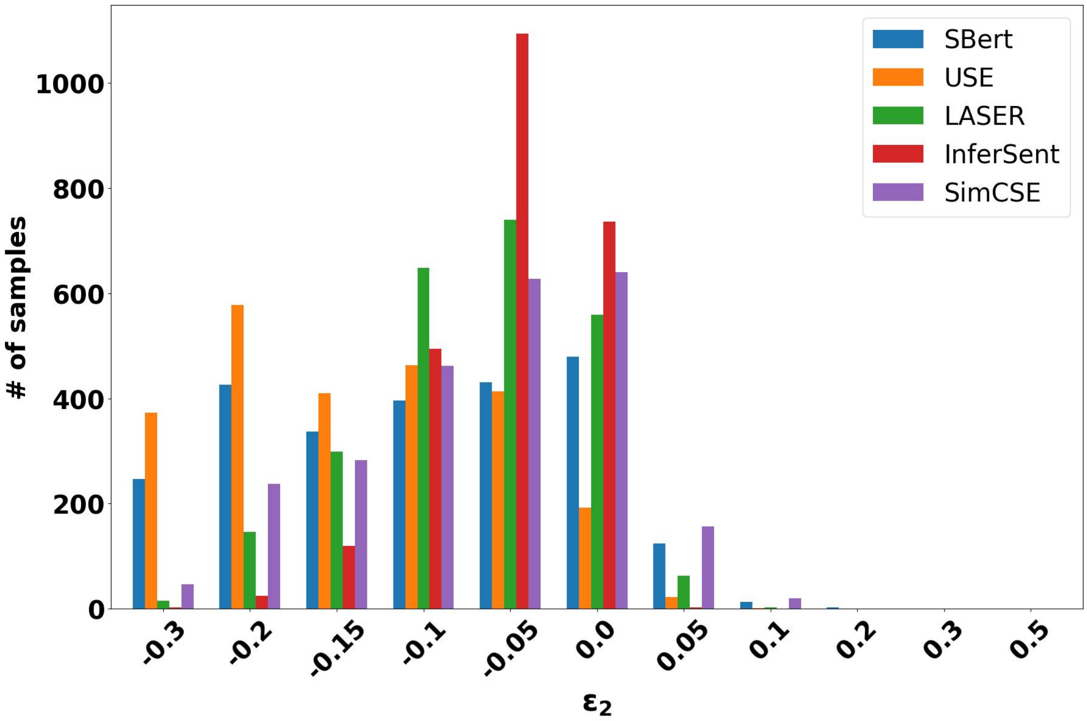
Figure 8: **Criterion-3**: The figure of **MRPC** dataset showcasing represent the difference in **NED distance** using the formula $NED(S, S'_A) - NED(S, S'_P) > \epsilon_{C3,N}$, where $\epsilon_{C3,N}$ denotes the expected minimum margin of differentiation. Note, that positive side represent better model.



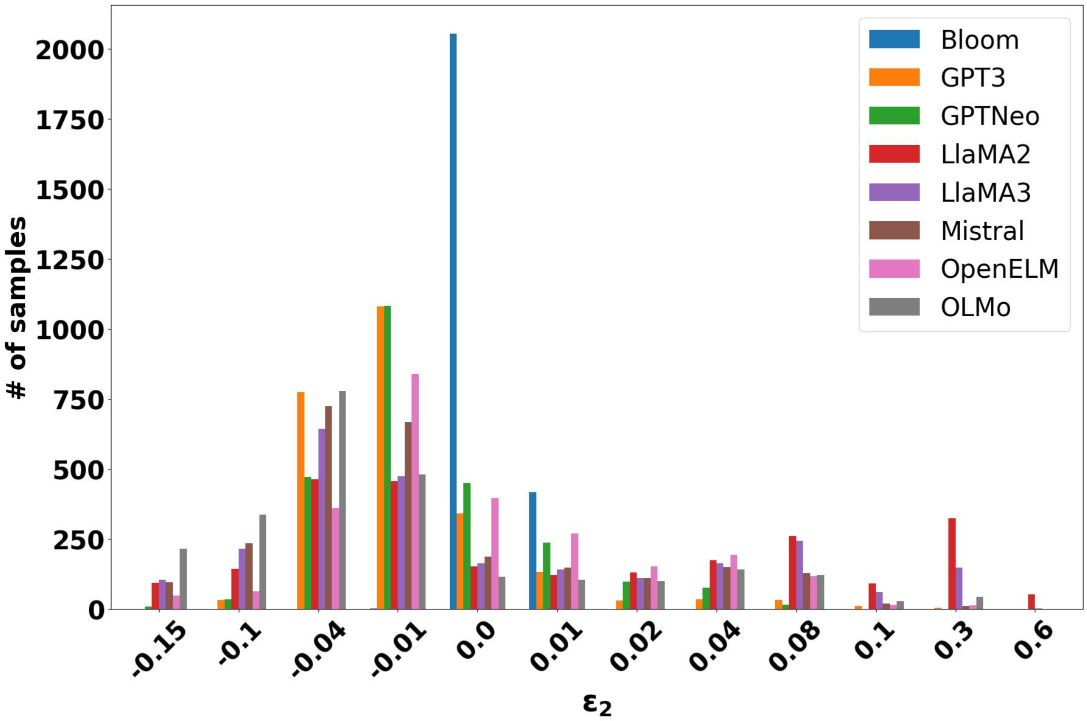
(a) **Classical Model** - **NED Metric** on **Antonym Replacement** Task on **PAW-WIKI** dataset

(b) **LLM** - **NED Metric** on **Antonym** Replacement Task on **PAW-WIKI** dataset

Figure 9: **Criterion-3**: The figure of **PAWS-WIKI** dataset showcasing represent the difference in **NED distance** using the formula $NED(S, S'_A) - NED(S, S'_P) > \epsilon_{C3,N}$, where $\epsilon_{C3,N}$ denotes the expected minimum margin of differentiation. Note, that positive side represent better model. Figure (a) Classical Encoders and (b) LLMs. It highlights their ability to distinguish between a sentence and its antonym counterpart on PAWS-WIKI.

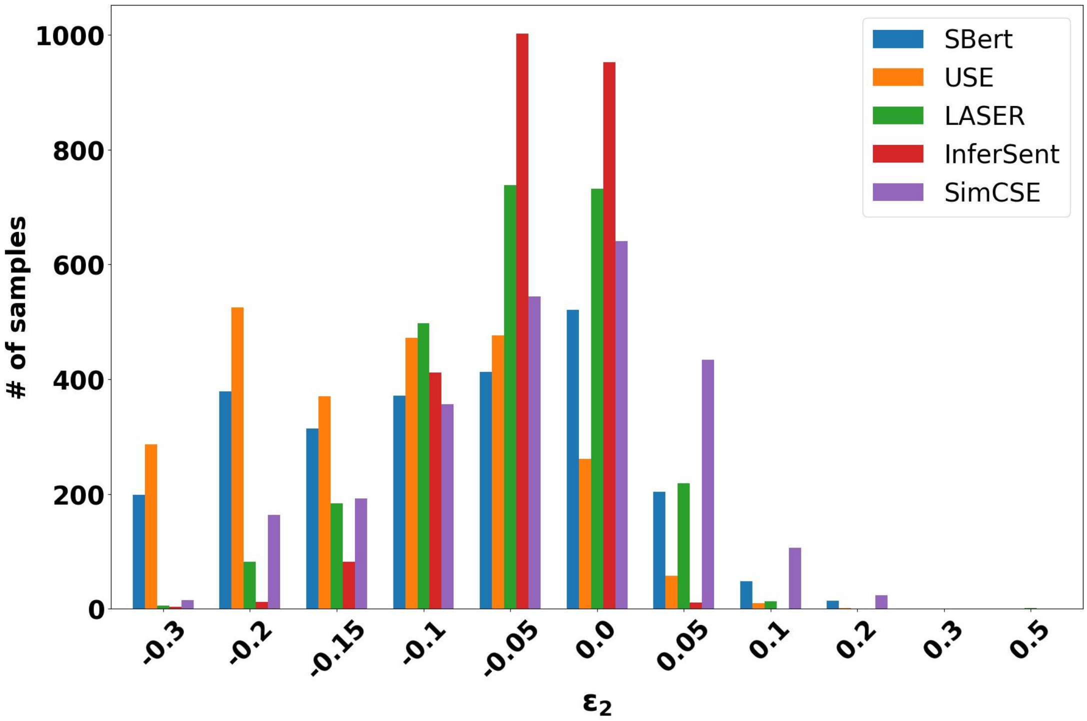


(a) Classical Model - **Sentence Jumbling** Task on **MRPC** dataset with **n=1**.

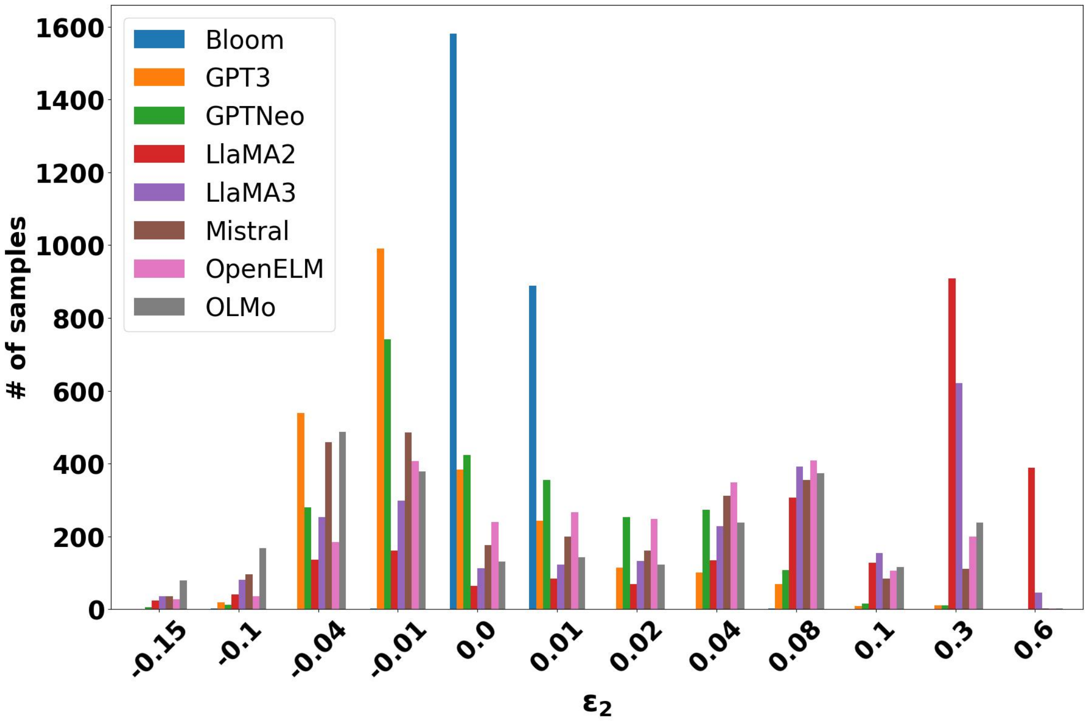


(b) LLMs - **Sentence Jumbling** Task on **MRPC** dataset with **n=1**.

Figure 10: **Criterion-5**: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for MRPC dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=1** on MRPC. The scores are computed using the formula $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.

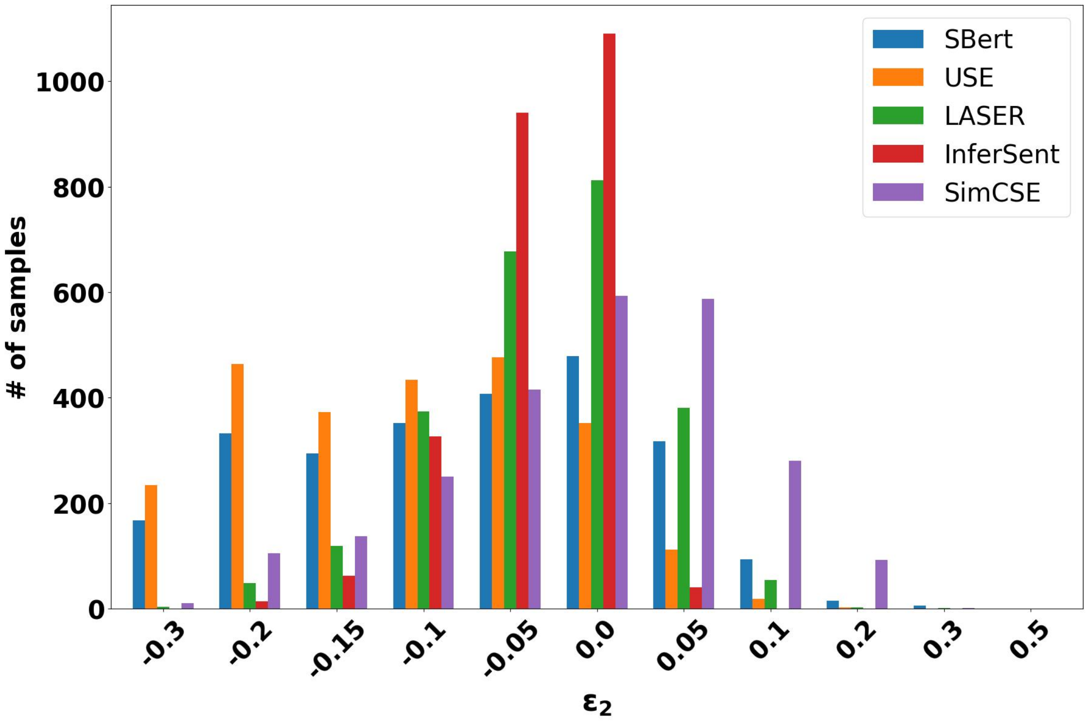


(a) Classical Model - **Sentence Jumbling** Task on **MRPC** dataset with **n=2**.

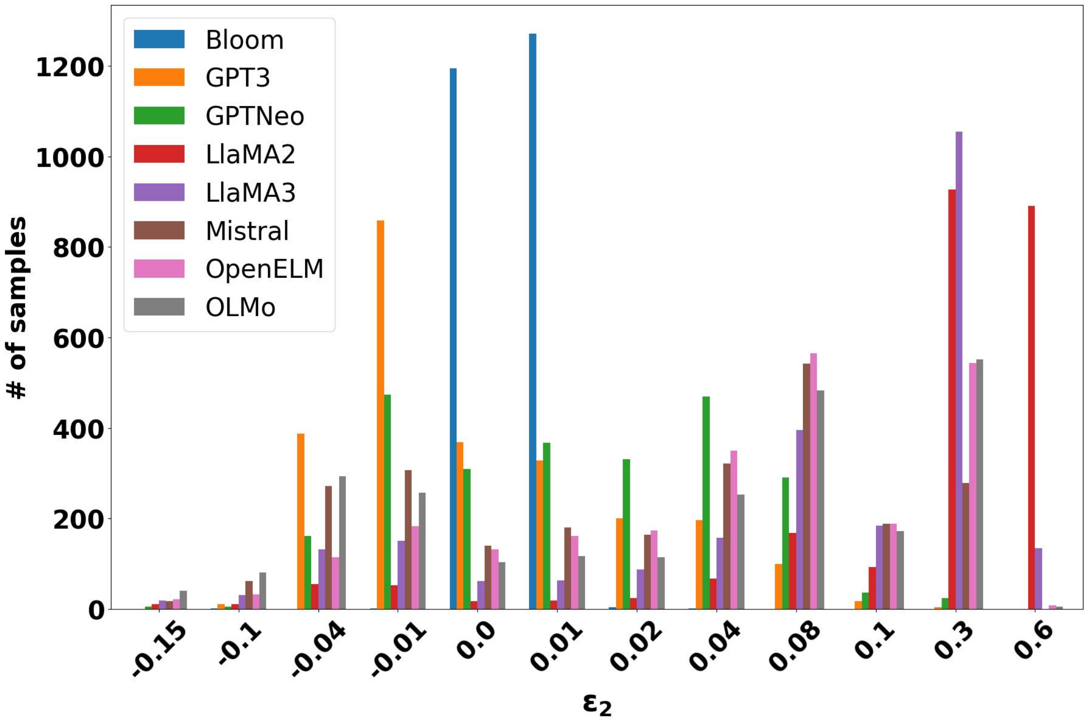


(b) LLMs - **Sentence Jumbling** Task on **MRPC** dataset with **n=2**.

Figure 11: **Criterion-5**: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for the MRPC dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=2** on MRPC. The scores are computed using the formula $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.

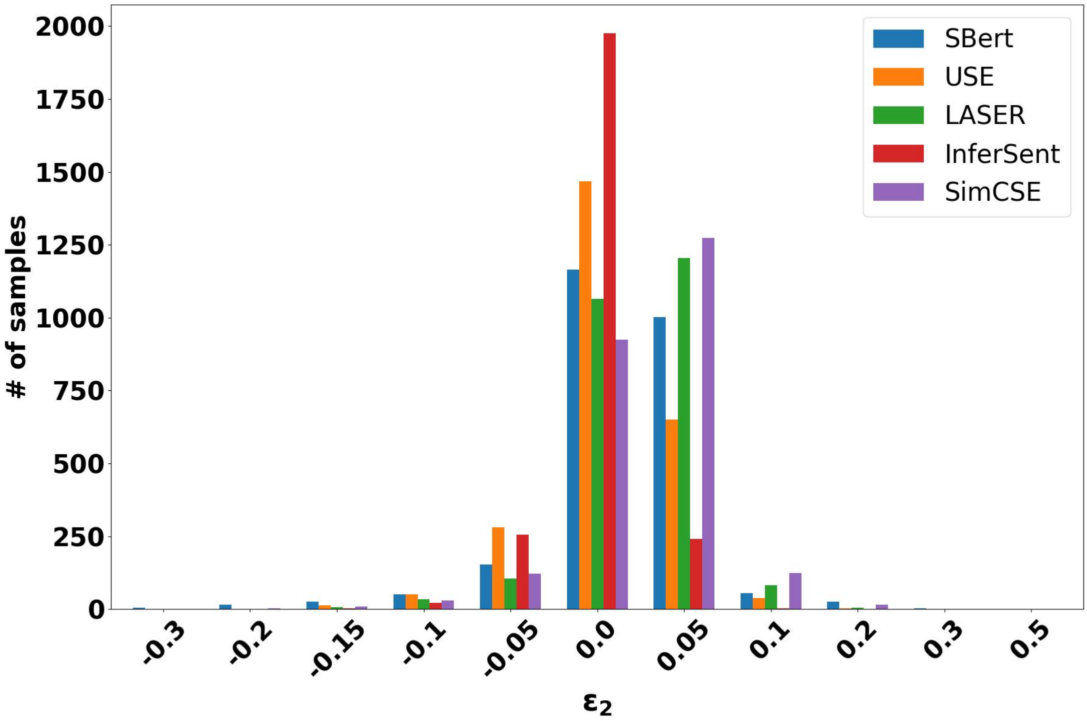


(a) Classical Model - **Sentence Jumbling** Task on **MRPC** dataset with **n=3**.

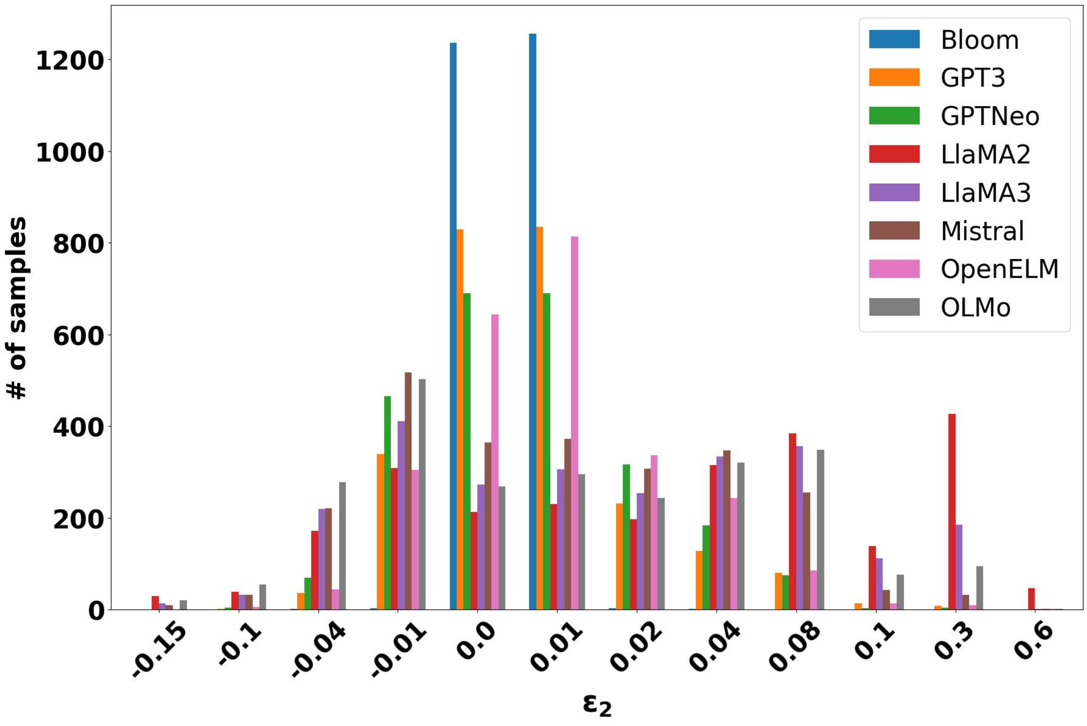


(b) LLMs - **Sentence Jumbling** Task on **MRPC** dataset with **n=3**.

Figure 12: **Criterion-5**: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for MRPC dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=3** on MRPC. The scores are computed using the formula $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.

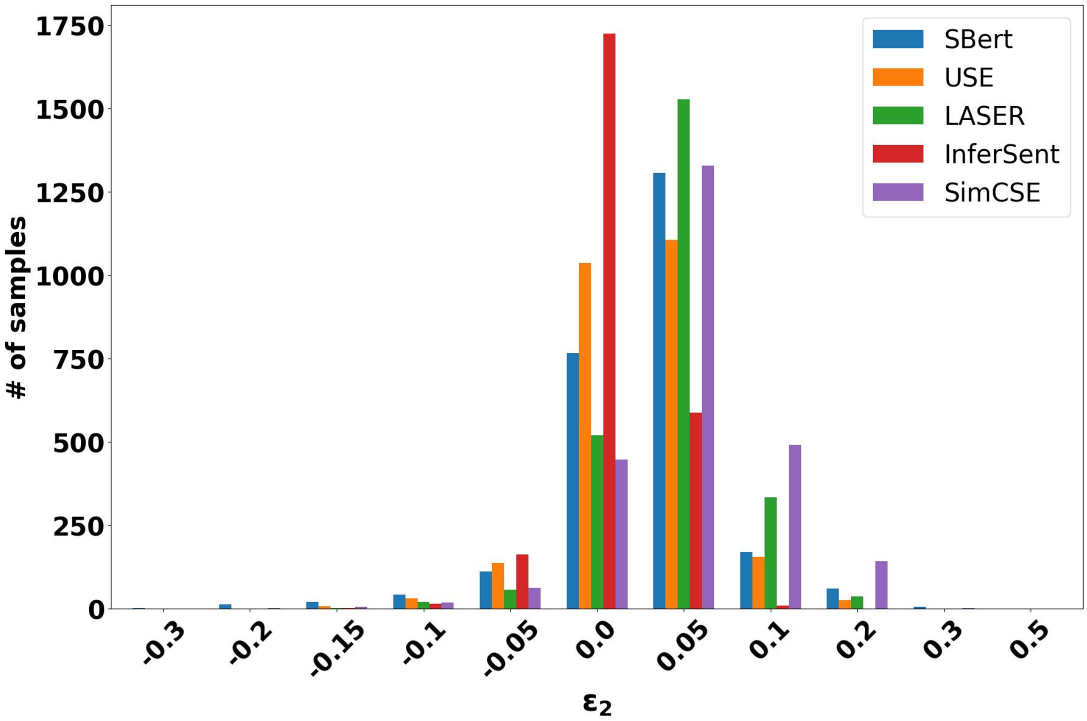


(a) Classical Model - **Sentence Jumbling** Task on **PAWS-WIKI** dataset with **n=1**.

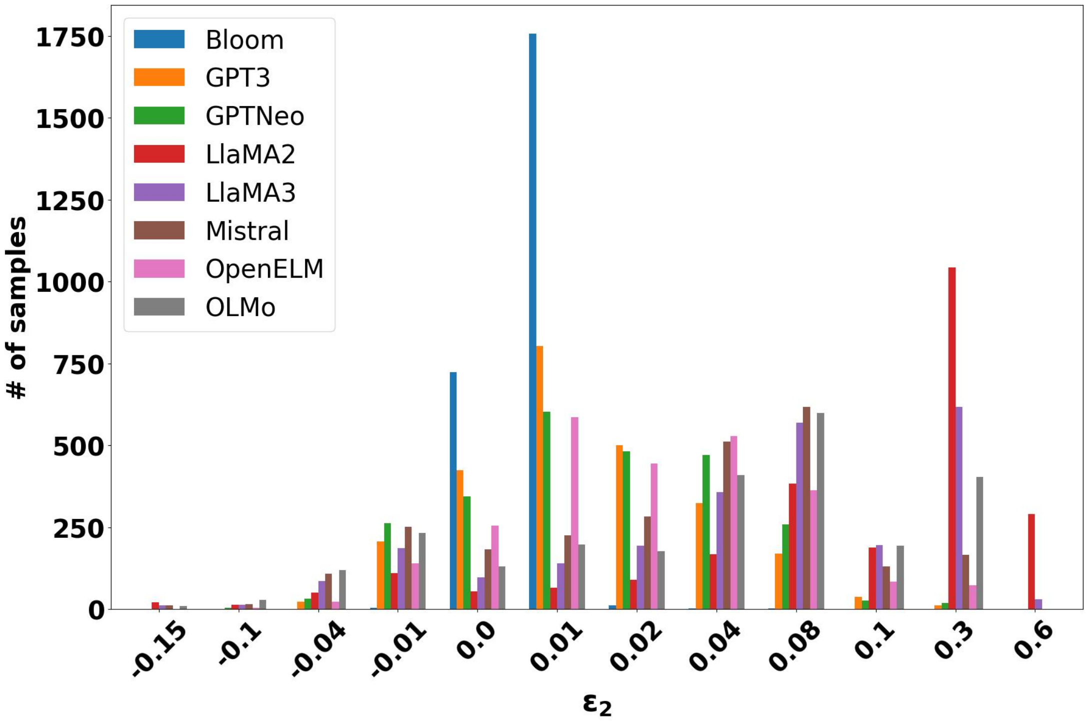


(b) LLMs - **Sentence Jumbling** Task on **PAWS-WIKI** dataset with **n=1**.

Figure 13: **Criterion-5**: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for PAW-WIKI dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=1** on PAWS-WIKI. The scores are computed using the formula $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.

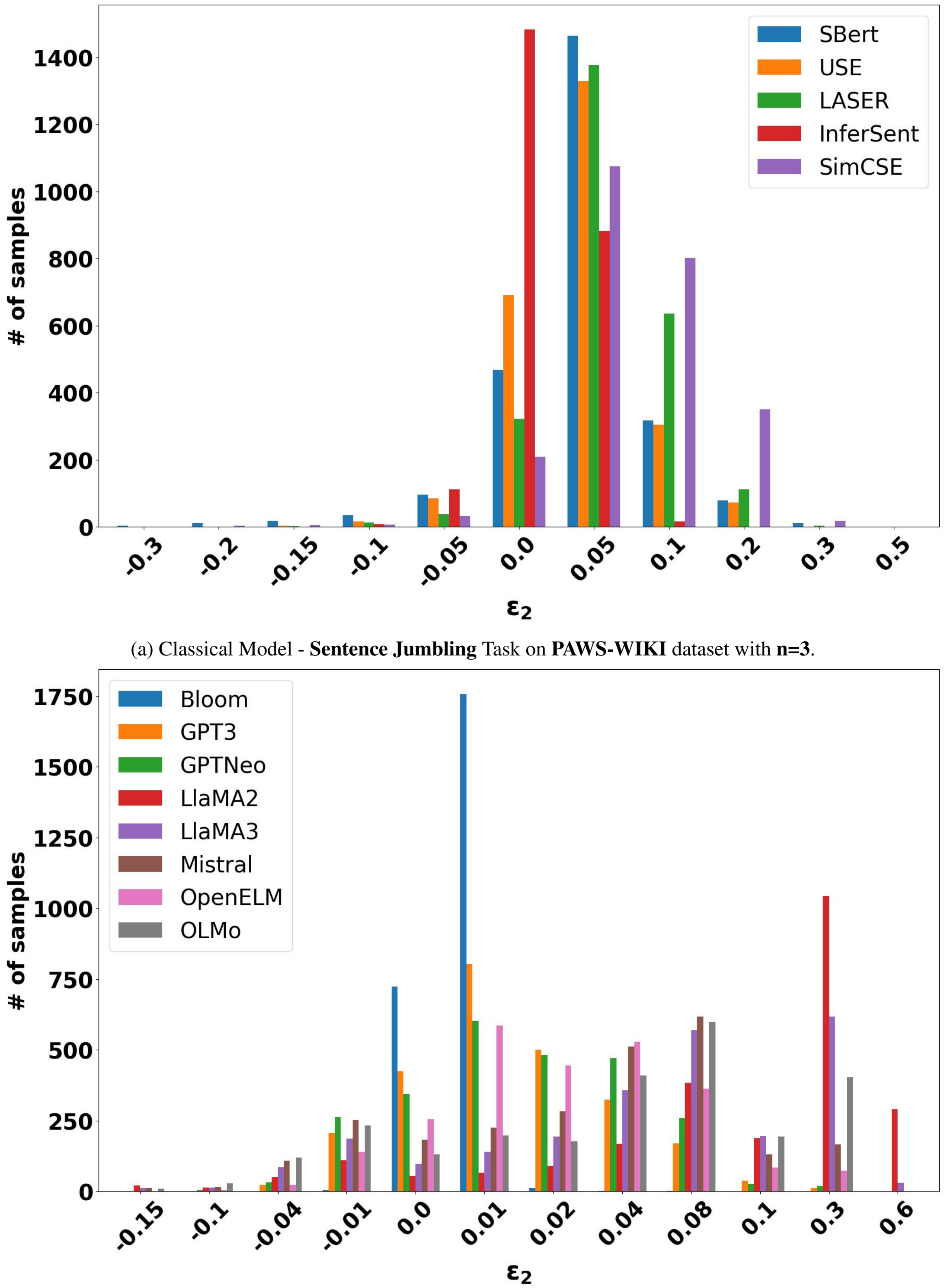


(a) Classical Model - **Sentence Jumbling** Task on **PAWS-WIKI** dataset with **n=2**.



(b) LLMs - **Sentence Jumbling** Task on **PAWS-WIKI** dataset with **n=2**.

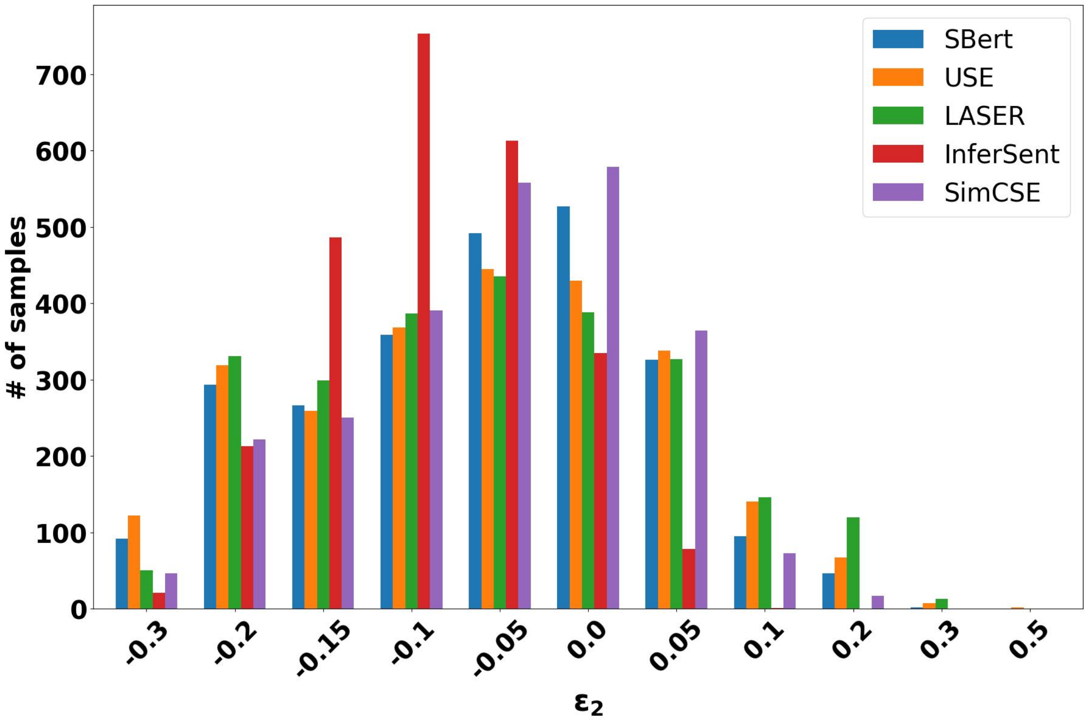
Figure 14: **Criterion-5**: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for PAW-WIKI dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=2** on PAWS-WIKI. The scores are computed using the formula $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.



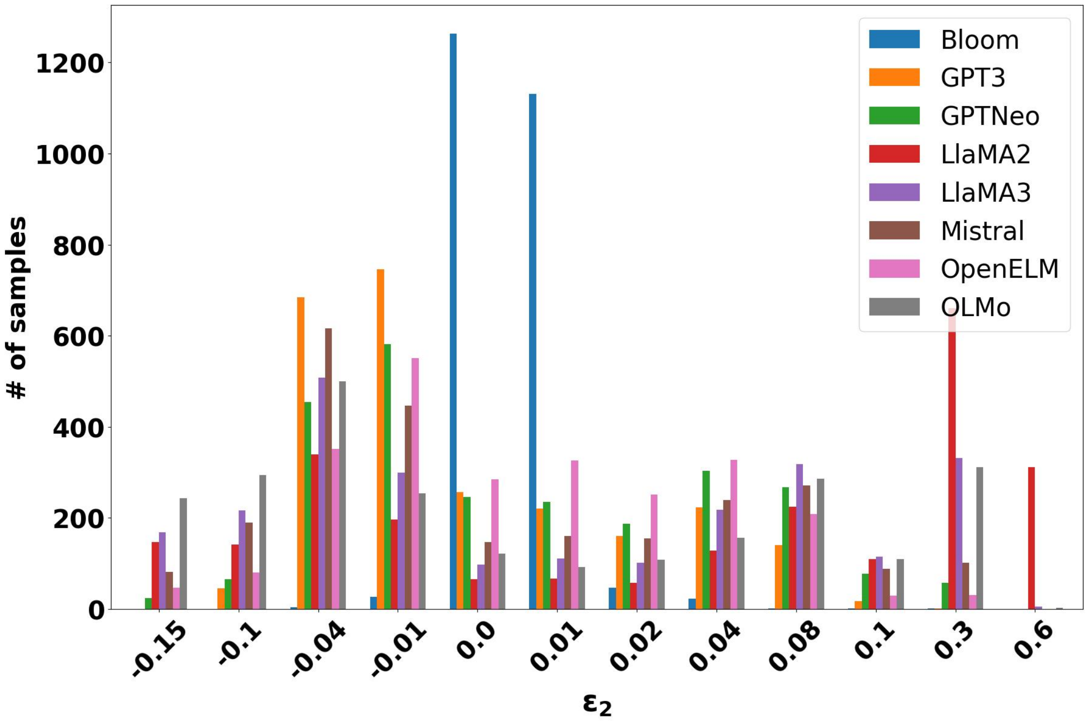
(a) Classical Model - **Sentence Jumbling** Task on **PAWS-WIKI** dataset with **n=3**.

(b) LLMs - **Sentence Jumbling** Task on **PAWS-WIKI** dataset with **n=3**.

Figure 15: **Criterion-5**: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for the PAW-WIKI dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=3** on PAWS-WIKI. The scores are computed using the formula $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.

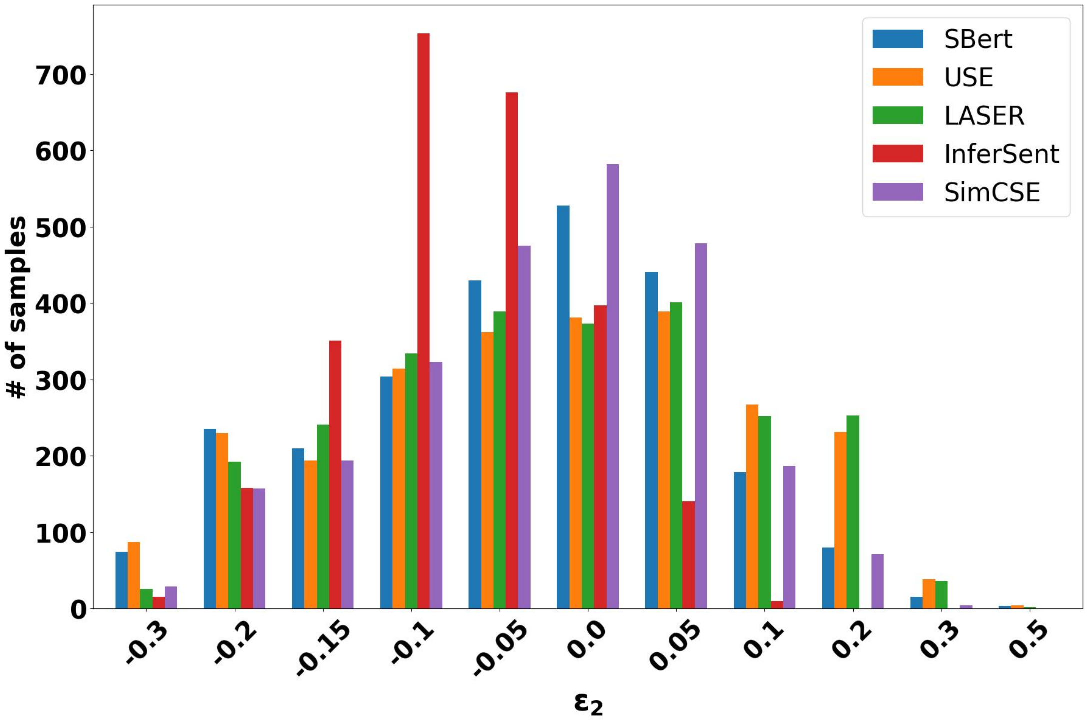


(a) Classical Model - **Sentence Jumbling** Task on **QQP** dataset with **n=1**.

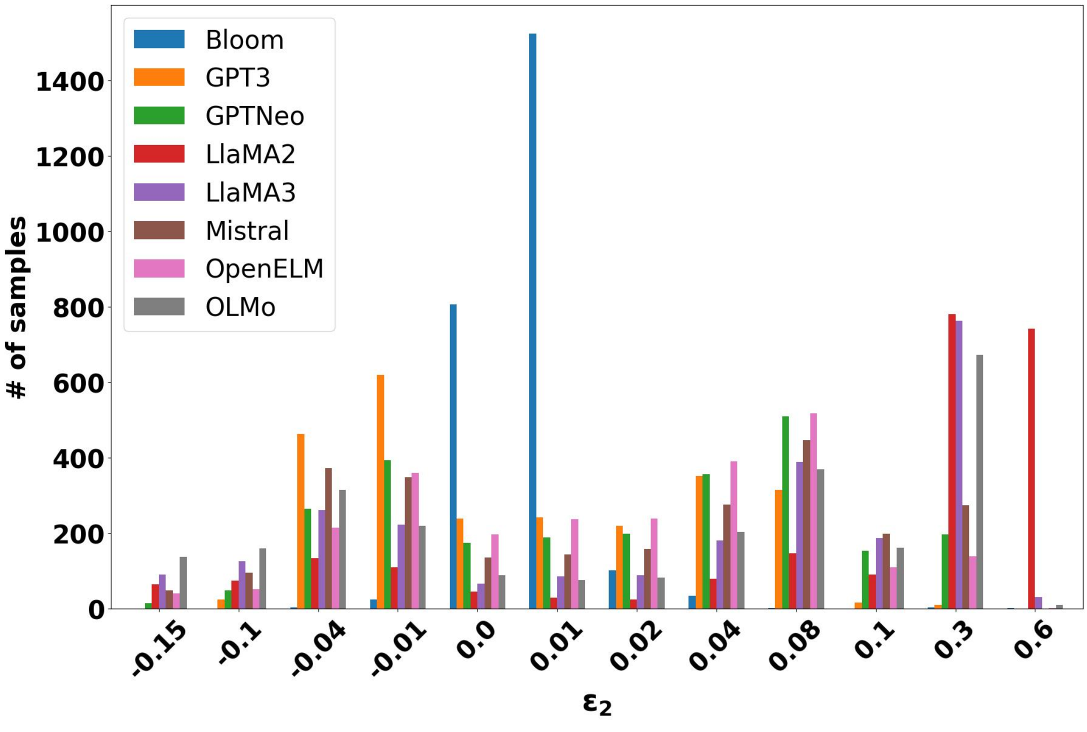


(b) LLMs - **Sentence Jumbling** Task on **QQP** dataset with **n=1**.

Figure 16: **Criterion-5**: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for QQP dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=1** on QQP. The scores are computed using the formula $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.

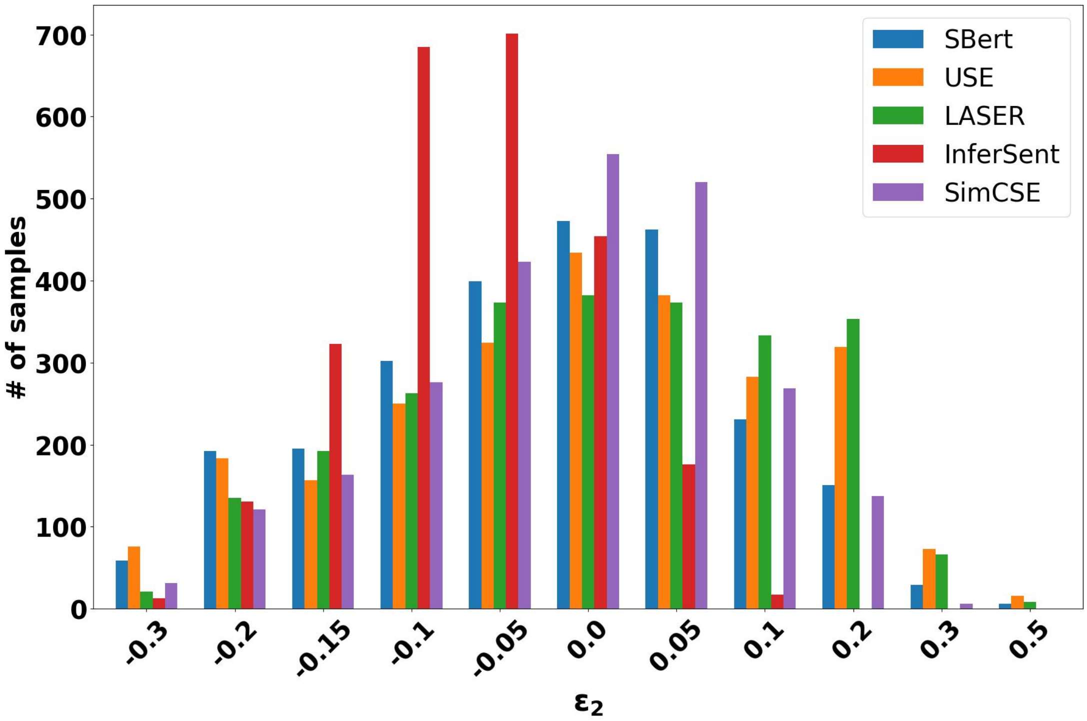


(a) Classical Model - **Sentence Jumbling** Task on **QQP** dataset with **n=2**.

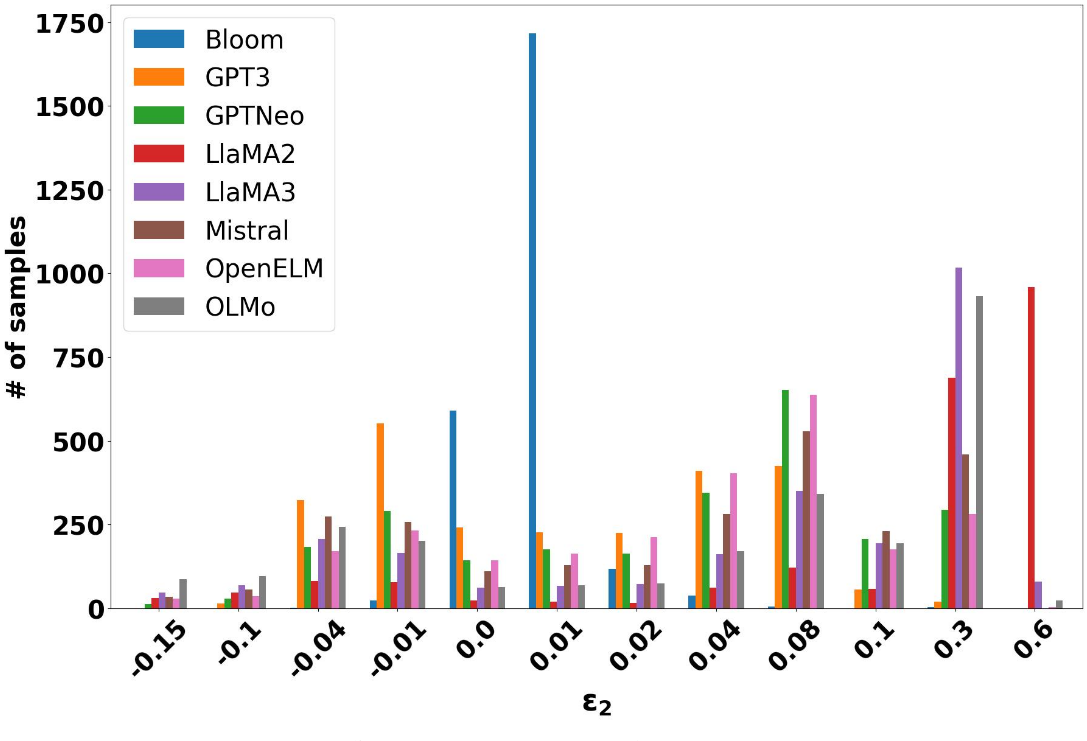


(b) LLMs - **Sentence Jumbling** Task on **QQP** dataset with **n=2**.

Figure 17: **Criterion-5**: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for QQP dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=2** on QQP. The scores are computed using the formula $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.

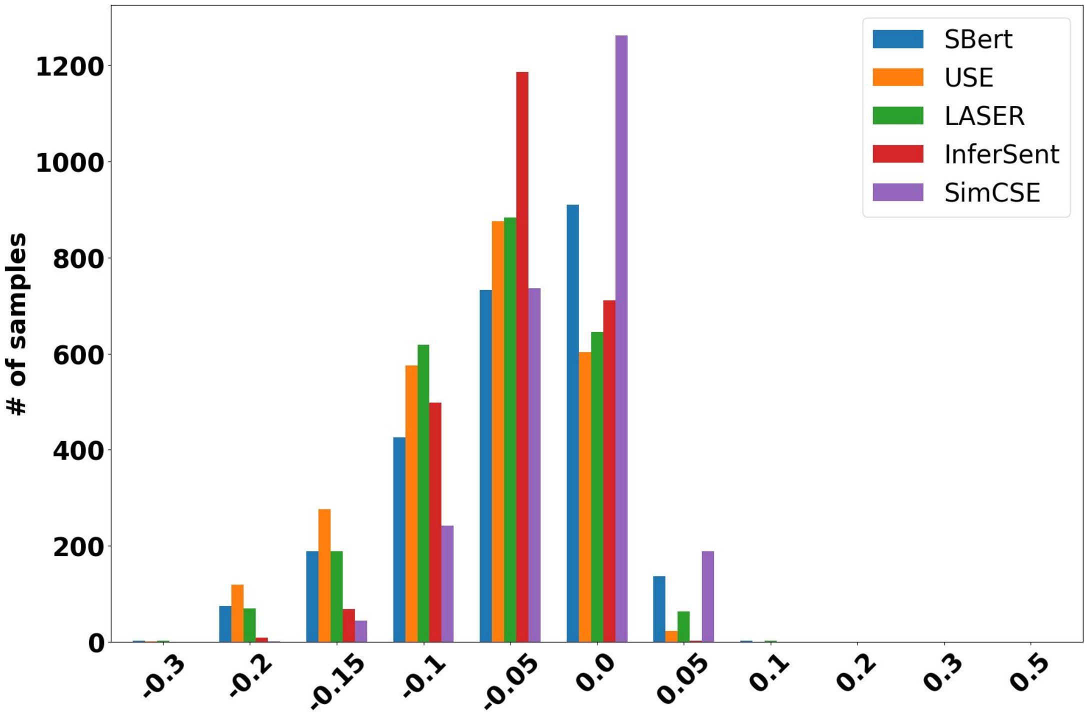


(a) Classical Model - **Sentence Jumbling** Task on **QQP** dataset with **n=3**.

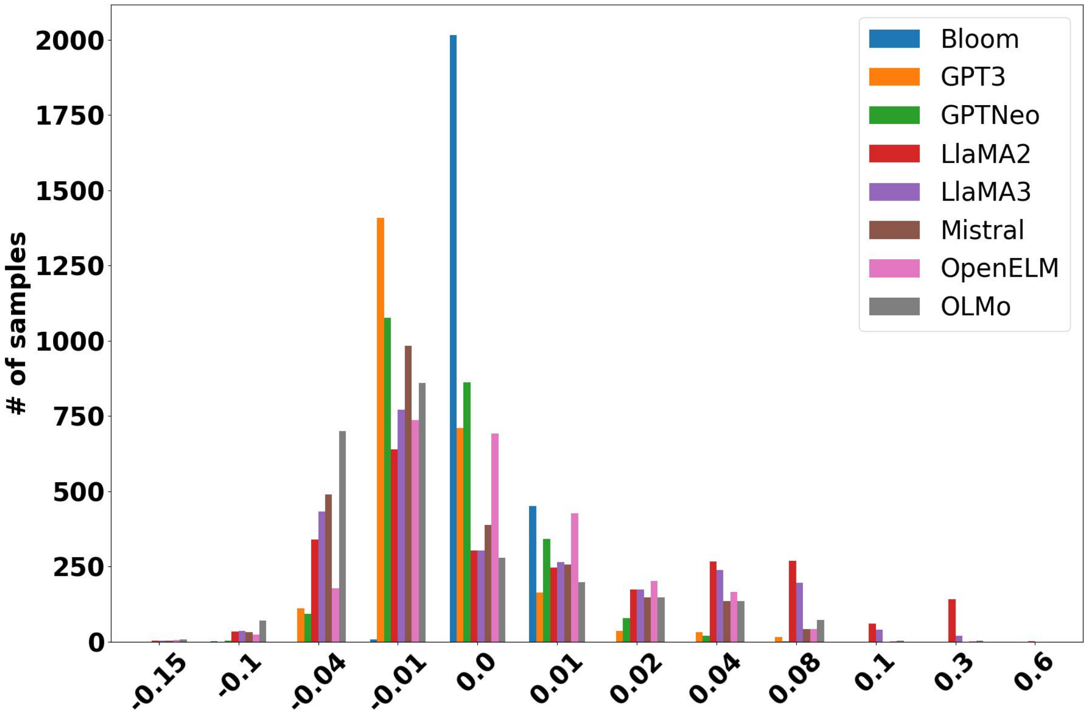


(b) LLMs - **Sentence Jumbling** Task on **QQP** dataset with **n=3**.

Figure 18: **Criterion-5**: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for QQP dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=3** on QQP. The scores are computed using the formula $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_{C5,S}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.

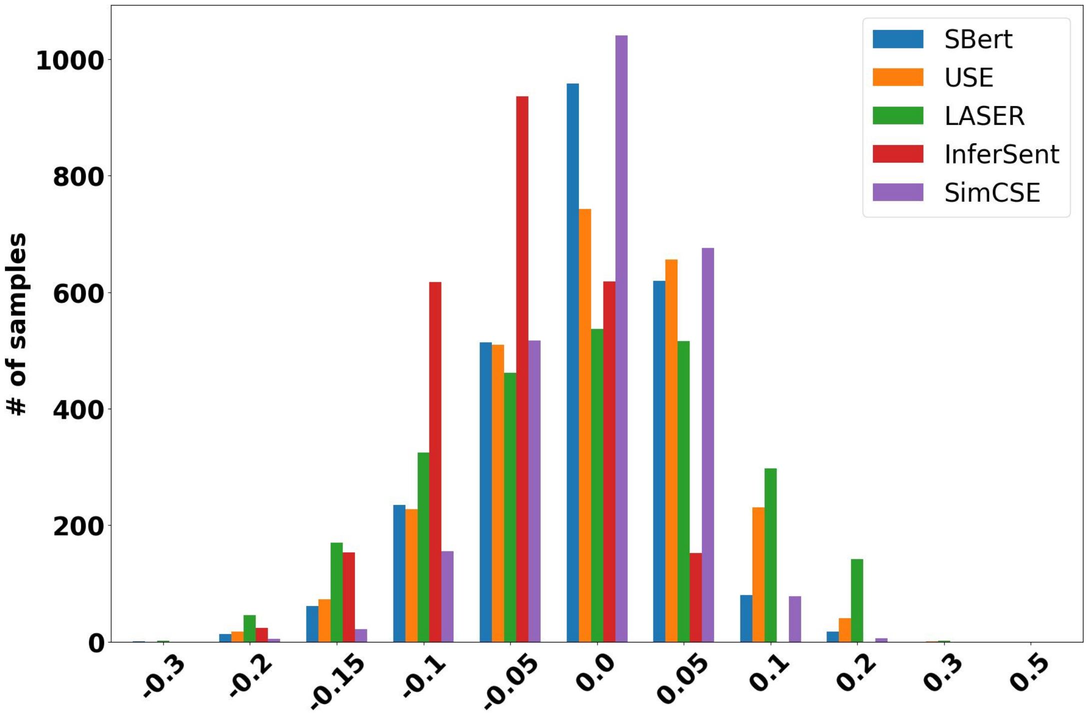


(a) Classical Model - **Sentence Jumbling** Task on **MRPC** dataset with **n=1**.

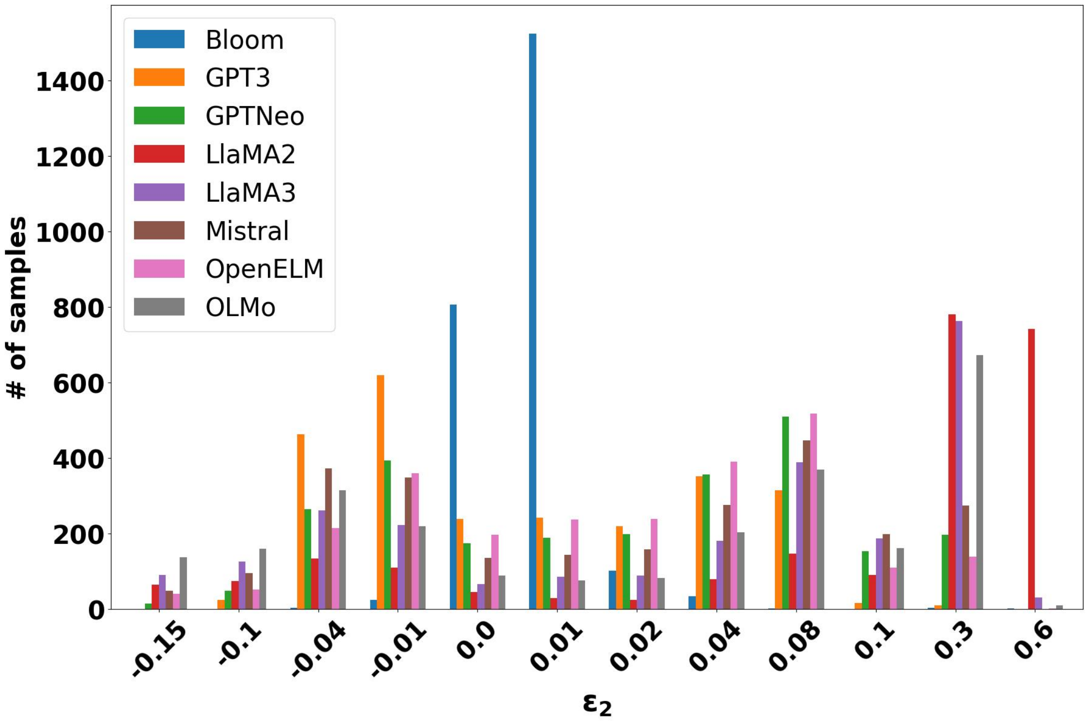


(b) LLMs - **Sentence Jumbling** Task on **MRPC** dataset with **n=1**.

Figure 19: **Criterion-5** The presented figures illustrate the results for the Jumble Sentence Criterion-5 for MRPC dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=1** on MRPC. The scores are computed using the formula $NED(S, S'_J) - NED(S, S'_P) > \epsilon_{C5,N}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.

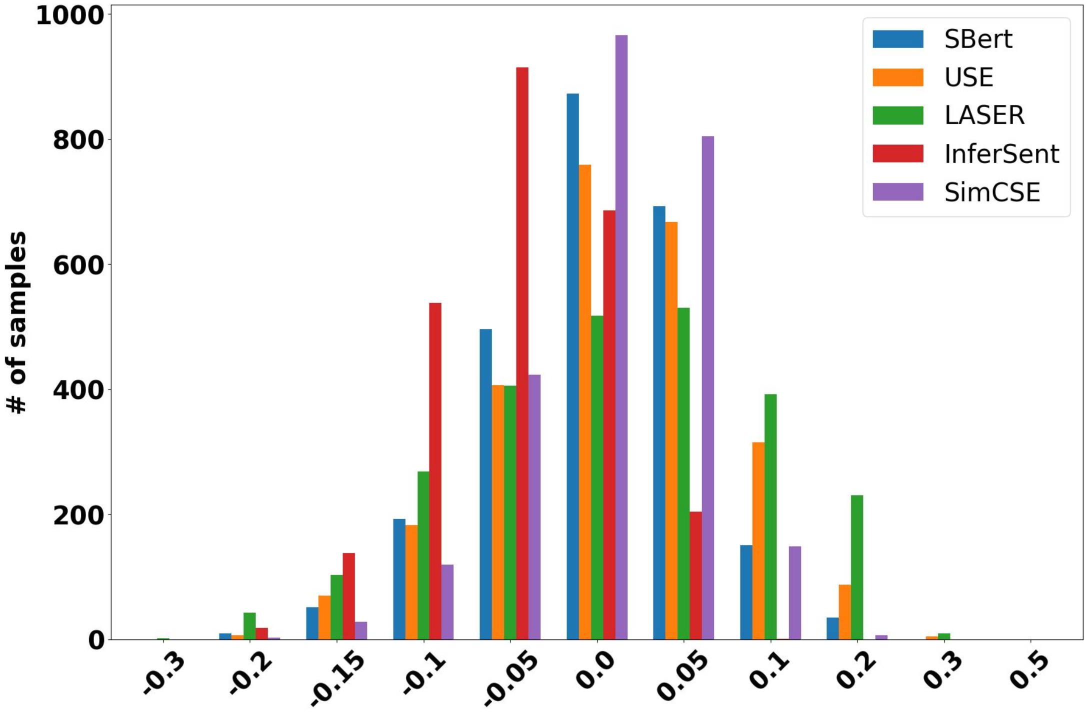


(a) Classical Model - **Sentence Jumbling** Task on **QQP** dataset with **n=2**.

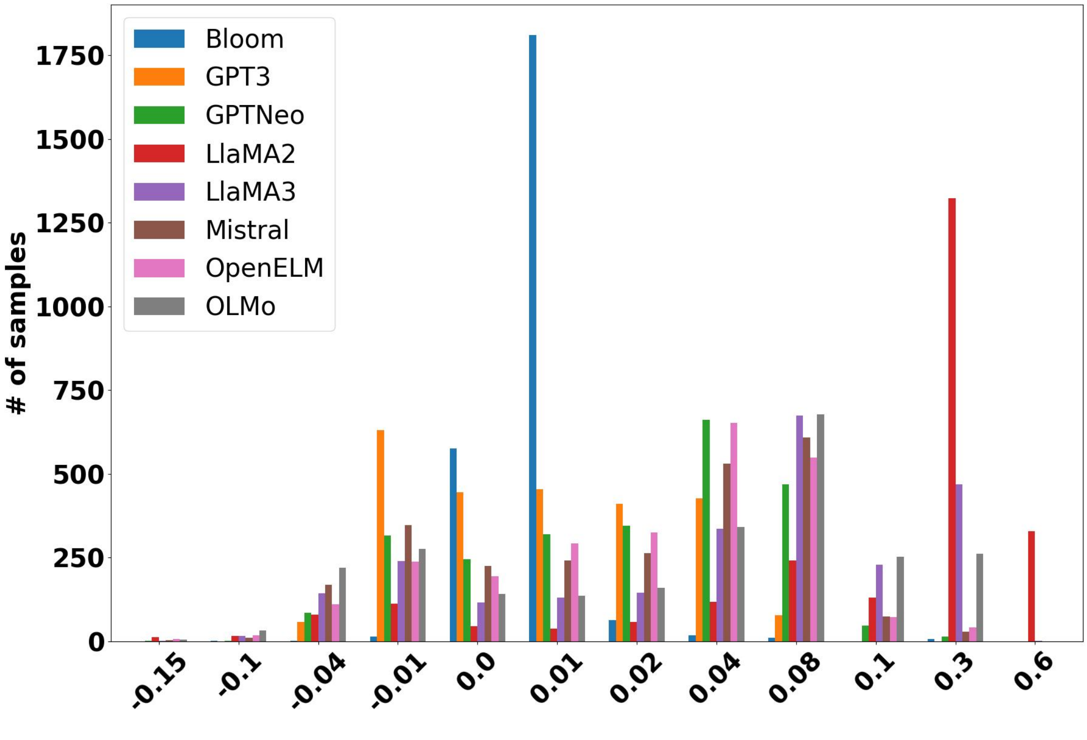


(b) LLMs - **Sentence Jumbling** Task on **QQP** dataset with **n=2**

Figure 20: **Criterion-5**: The presented figures illustrate the results for the Jumble Sentence for QQP dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=2** on QQP. The scores are computed using the formula $NED(S, S'_J) - NED(S, S'_P) > \epsilon_{C5,N}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.



(a) Classical Model - **Sentence Jumbling** Task on **QQP** dataset with **n=3**.



(b) LLMs - **Sentence Jumbling** Task on **QQP** dataset with **n=3**.

Figure 21: **Criterion-5**:The presented figures illustrate the results for the Jumble Sentence for QQP dataset. Figures (a) and (b) depict histograms for classical and llms, respectively, highlighting their ability to distinguish between a sentence and its jumbled counterpart when the order of jumbling is **n=3** on QQP. The scores are computed using the formula $NED(S, S'_J) - NED(S, S'_P) > \epsilon_{C5,N}$ denotes the expected minimum margin of differentiation. The x-axis quantifies the range of scores, with each bin signifying the aggregate of data points falling within that specific range. Conversely, the y-axis enumerates the number of samples populating each bin.