The Daunting Dilemma with Sentence Encoders: Glowing on Standard Benchmarks, Struggling with Capturing Basic Semantic Properties

Yash Mahajan
Auburn University
yzm0034@auburn.edu

Naman Bansal
Auburn University
nbansal@auburn.edu

Shubhra Kanti Karmaker ("Santu")
Auburn University
sks0086@auburn.edu

Abstract

In recent years, the latest LLMs like GPT3 and LlaMa2 have impressed in various fields. However, comparing them with classical models like SBert and USE on the SentEval benchmark reveals an interesting trend. Despite being big and needing a lot of computing power, the latest LLMs perform similarly to the classical models which are more resource-friendly. To explore deeper, we evaluated sentence encoders on five proposed criteria: *paraphrasing, synonyms replacement, antonyms replacement, paraphrasing without negation, and sentence jumbling*. Except for LlaMa2, all models struggled with antonym replacement and sentence jumbling. These findings emphasize that although LLMs have come a long way, they still lack some basic meanings. This highlights the importance of more rigorous benchmarks as well.

1 Introduction

Transformer-based Language Models (LLMs) like SBERT (Reimers and Gurevych, 2019), GPT-3 (Brown et al., 2020), and LLaMA (Touvron et al., 2023) have profoundly reshaped the NLP domain, consistently demonstrating enhanced performance across standard benchmarks and key downstream tasks, including translation, question answering, and text classification. One prominent application of these models is the generation of *"sentence embeddings"*, which serve as proxy representations of sentences. However, achieving robust and stable sentence representations remains an unresolved challenge. Additionally, the capacity of these models to grasp fundamental linguistic properties is yet to be clearly established (Pham et al., 2021).

To investigate this in detail, we adopt a retrospective approach and compare the performance of popular nine-sentence encoders (refer appendix A.3) over the proposed five criteria. The criteria are 1) Paraphrasing, 2) Synonym Replacement, 3) Antonym Replacement, 4) Paraphrasing without Negation, and 5) Sentence Jumbling. We use these criteria to quantify how well a sentence encoder can capture the semantic similarity between two sentences. Note that these criteria only constitute a subset of linguistic properties that we argue a good sentence encoder should hold, *but it is far from an exhaustive list, which is beyond the scope of this paper*. Furthermore, these criteria can be experimentally evaluated in an unsupervised fashion without requiring a specific downstream task.

2 Semantic Evaluation Criteria

To study the basic linguistic understanding of sentence encoders/language models, we devise the following evaluation criteria.

Criterion-1 (Paraphrasing): We argue that "*A good sentence encoder should create similar embeddings for paraphrased sentences and significantly different ones for unrelated sentences*".We measure this by computing the difference between average similarity scores for paraphrases and non-paraphrases. A good encoder is expected to show a larger difference.

Criterion-2 (Synonym Replacement): We argue that "If we replace n words (where n is small) from sentence S with their synonyms, a good sentence encoder will produce similar embeddings for the original and perturbed sentences S'_P." This is because synonym replacement doesn't change the meaning much. We measure this by computing the difference in average similarity scores. A good encoder is expected to show a larger difference.

Criterion-3 (Paraphrase Vs. Antonym Replacement) : We argue that when given a sentence S, its paraphrase S'_P, and an antonym-replaced sentence S'_A, the paraphrase should be more similar to S than the antonym-replaced one by a clear margin ($Sim(S, S'_P) - Sim(S, S'_A) > \epsilon_1$). This ensures a good sentence encoder can distinguish between

paraphrases and antonym-replaced sentences in the semantic space.

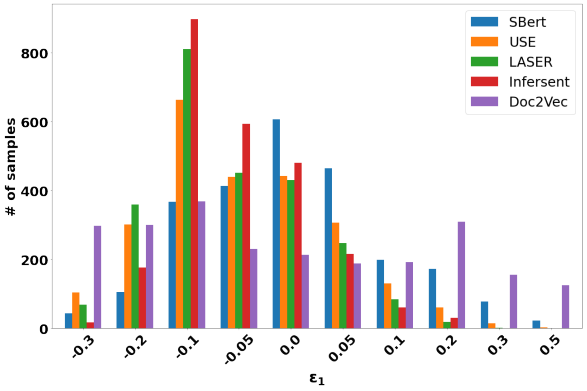
Criterion-4 (Paraphrasing without Negation) : In this criterion, we evaluate encoders understanding of negation. We took a sentence with negation S and a paraphrase of sentence S without negation, S'. Hence, making S' a affirmative representation of the sentence S (Hossain and Blanco, 2022). Next, we quantified the difference between the average cosine similarity scores of the pair of sentences. The intuition here is that a *"good"* sentence encoder will recognize the semantic equivalence despite negation being present in S but not in S', and thus produce high similarity scores.

Criterion-5 (Paraphrase Vs. Sentence Jumbling) : Instead of replacing words, we focus on jumbling the sentence. For a sentence S and its paraphrase S'_P, compared to a jumbled sentence S'_J, S'_P should be more similar to S by a clear margin ($Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_2$). This ensures a good sentence encoder creates embeddings where any paraphrase is closer to the original sentence than a jumbled one in the semantic space.

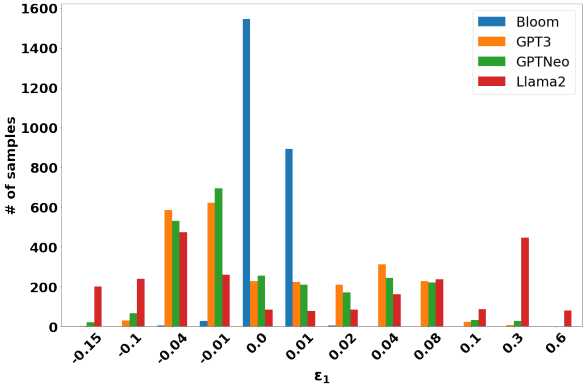
3 Results and Conclusion

To measure similarity, we calculated the average cosine similarity for each criterion. For criteria 1, 2, and 4, we adjusted scores to remove biases by normalizing with random pairs.

When comparing all models on the SentEval benchmark (refer appendix 1), we found both classical and emergent models performed competitively. GPT3-ada had the highest average accuracy at 90.23%, followed by SBert (86.90%) and LlaMa2 (86.48%). Moving on to criteria comparisons. In criterion 1, SBert stood out by outperforming the latest models like GPT3-ADA and LlaMA. The same trend was observed in criterion 4, which is similar to criterion 1, except one sentence has negation. SBert excelled, demonstrating its good semantic understanding. In criterion 2, where we expect similar results to criterion 1, SBert again outperformed all other models, followed by GPT3-ada. A potential reason for the latest models' underperformance could be their design, mainly tailored for text generation, possibly making them better for longer sequences than shorter ones. Notably, all models struggled with the WIKI dataset, showing confusion due to high lexical overlap between sentence pairs. In criterion 3, models, except LlaMa-2, couldn't distinguish between antonym sentences (refer to figure 1). LlaMa-2 performed better, while



(a) Classical Model - Antonym Replacement on QQP



(b) Emergent Model - Antonym Replacement on QQP

Figure 1: The figure shows cosine similarity variations between Paraphrased and Antonym hypotheses, determined by $Sim(S, S'_P) - Sim(S, S'_A) > \epsilon_1$. Bins on the x-axis group data, each bin indicating samples within a specific ϵ_1 range. This figure pertains to the QQP dataset, while figures for MRPC and PAWS-WIKI datasets are in Appendix A.4.

GPT3-ada and GPTNeo were suboptimal. The lack of exposure to enough negation or antonym sentence pairs during pre-training could be a reason for the encoders' struggle. In criterion 5, after swapping one word (i.e. $n = 1$), most models failed to capture the impact of jumbled words on similarity. As the word swapping increased, all models continued to struggle, except for LlaMa2, which showed improvement. Among all models, Bloom, LASER, Infersent, and D2v faced challenges in all criteria, while SBert excelled in criteria 1, 2, and 4. LlaMa2 performed better in criteria 3 and 5. Further details can be found in the appendix.

To conclude, a good sentence encoder should pass all criteria, and SBert demonstrated strong performance in various aspects of semantic understanding, outshining other models in certain scenarios.

Limitations

Our findings are limited to the English language. The experiments are primarily focused on unsupervised semantic understanding tasks where no training data or previous observations about the goal task are available. Thus, evaluation of the constructed perturbed sentences is required. Therefore, our findings may not hold for all possible downstream NLP tasks. However, in the absence of available training data for a particular problem, our findings can still be useful in choosing a suitable sentence encoder and designing initial experiments.

References

Mikel Artetxe and Holger Schwenk. 2019. 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.

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.

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. 2017. 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.

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.

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.

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

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.

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

OpenAI. 2022. Gpt3-text embedding.

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.

Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. 2021. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1145–1160.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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, 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.

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*.

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

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

A Appendix

A.1 SentEval

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 or multi-layered perceptron (MLP), 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 five 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).

When comparing all models on the SentEval (Conneau and Kiela, 2018) benchmark. It is a widely used framework for evaluating the efficacy of sentence embeddings. The accuracy scores of each sentence encoder on SentEval can be found in Table 1. Results reveal a strong performance by large models, with GPT-3 achieving the highest average accuracy of 90.23% across datasets. However, small models like SBERT remain highly competitive (86.90%), underscoring their efficiency despite utilizing far fewer parameters, and even surpassing LlaMa2 and other large models. It is interesting that, SBert model correctly identifies the distinguishing paraphrase and non-paraphrase sentences in the MRPC dataset. It is probably due to its Siamese-like model architecture. Furthermore, the close proximity of USE and Infersent highlights the capabilities of smaller encoders. More crucially, there is no unique top performance across datasets, highlighting issues with generalizability - a key criterion for a good sentence encoder. For instance, GPT-3 and SBERT faltered on MRPC and TREC respectively, despite success on other benchmarks. These observations indicate that the models have limited contextual understanding. To deeply probe this phenomenon, we developed the five rigorous semantic tests described in Section 2.

A.2 Datasets

In this work, we used three publicly available paraphrasing datasets with human-annotated labels. All three datasets come with binary labels assigned to each pair of sentences. Label 1 (Pos) indicates that the pair of sentences have a similar meaning and 0 (Neg) indicates otherwise. The datasets are **1) QQP** (Quora Questions Pair) dataset (Chen et al., 2018), which is a collection of paraphrased and non-paraphrased pairs of questions. **2) PAWS-WIKI** (Paraphrase Adversaries from Word Scrambling-Wikipedia) dataset (Zhang et al., 2019), which is a collection of pair of sentences from Wikipedia with high lexical overlaps. And, **3) MRPC** (Microsoft Research Paraphrasing Corpus) dataset (Dolan and Brockett, 2005), which is a collection of sentence pairs extracted from news articles. We also experimented with **Afin** dataset (Hossain and Blanco, 2022) which contains sentences with negations and their paraphrases without negation, representing challenging paraphrase examples.

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
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
Doc2Vec	49.76	63.76	49.16	68.77	49.92	19.20	66.49	52.43
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
GPT3	**88.36**	**93.08**	95.31	**91.29**	**93.63**	**96.00**	73.97	**90.23**

Table 1: 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.

A.3 Models

In total we evaluate 9 LLM models. 1) Universal Sentence Encoder (USE) (Cer et al., 2018), 2) Sentence-BERT (SBert) (Reimers and Gurevych, 2019), 3) InferSent (Conneau et al., 2017), 4) Language-Agnostic-SEntence Representation (LASER) (Artetxe and Schwenk, 2019), and 5) Document To Vector (Doc2Vec or D2V) (Le and Mikolov, 2014). 6) GPT3-Ada[1] (OpenAI, 2022), 7) LlaMa2 (Touvron et al., 2023), 8) Bloom (Scao et al., 2022), and 9) GPTNeo (EleutherAI, 2023).

A.4 Results

A.4.1 Criteria 1

Table 2 displays criterion-1 results. Classical SBERT excelled, distinguishing paraphrased and non-paraphrased sentences across all datasets, followed by USE. Emerging models like GPT3, GPT-Neo, and LlaMa2 showed sub-optimal performance, with Bloom and D2V failing to differentiate. Potentially, emerging models, designed for text generation, struggle with shorter sequences. All models performed poorly on WIKI due to high lexical overlap, confusing encoders. Comparing with SentEval results (Table 1) revealed discrepancies, questioning emergent models' effectiveness in sentence-level semantics.

A.4.2 Criteria 2

To generate synonym-perturbed sentences, we randomly selected n ($n = 1, 2, 3$) verbs or adjectives and replaced them with synonyms from the *WordNET* toolkit (Miller, 1995). This perturbation ensured high lexical overlap, distinct from Criterion-1. After this replacement, both original and perturbed sentences were encoded using various sentence encoders (refer to Section A.3), and their cosine similarities were computed. Normalized average similarity scores were calculated and reported in Table 3 for all three datasets. SBERT and GPT3 consistently exhibited high similarity scores, with USE closely following. SBERT excelled when $n = 1$ in QQP and MRPC, while GPT3 outperformed as n increased, showcasing its ability to handle greater sentence variation. Comparing this criterion with the SentEval benchmark in Table **??** revealed a consistent trend: SBERT and GPT3 excelled, while LlaMa2 and GPTNeo fell short. The Bloom model struggled similarly in both scenarios, showcasing deficits in distinguishing sentences. In conclusion, classical models like SBERT and USE, alongside the emerging GPT3, effectively capture semantic nuances, while many other models struggle, highlighting the competitiveness of classical models with smaller sizes compared to emergent counterparts.

A.4.3 Criteria 3

Figures 2 and 3 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 emergent encoder models also face challenges, the LlaMa2 model evidences a modest edge over its classical counterparts. Surprisingly, the D2V—a classical encoder—manifests good performance. This unexpected behavior warrants further exploration to derive a conclusive ex-

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

Model		USE	SBERT	Infer-Sent	LASER	D2V	Bloom	GPTNeo	GPT3-Ada	LlaMa-2
QQP	Pos	0.7553	0.8526	0.3182	0.3652	0.2516	0.0059	0.2669	0.2609	0.4277
	Neg	0.5278	0.5488	0.2849	0.3124	0.2368	0.0059	0.2512	0.2367	0.3734
	Diff	**0.2275**	**0.3038**	0.0333	0.0528	0.0148	0.0001	0.0157	0.0242	0.0543
WIKI	Pos	0.8645	0.9506	0.3552	0.4268	0.5180	0.0059	0.2767	0.2719	0.4646
	Neg	0.8554	0.9408	0.3552	0.4136	0.5402	0.0059	0.2750	0.2703	0.4568
	Diff	**0.0091**	**0.0098**	0.0000	0.0132	-0.0222	0.0000	0.0016	0.0016	0.0077
MRPC	Pos	0.7098	0.8134	0.3367	0.3828	0.4440	0.0059	0.2706	0.2634	0.4442
	Neg	0.6097	0.5488	0.3256	0.3564	0.3700	0.0059	0.2652	0.2549	0.4243
	Diff	**0.1001**	**0.2646**	0.0111	0.0264	0.0740	0.0001	0.0053	0.0085	0.0198

Table 2: Normalized Average Cosine Similarity for Criterion-1 (Paraphrasing task). Here, Positive (**Pos.**) and Negative (**Neg.**) means paraphrase-pairs and non-paraphrase-pairs, respectively. **Diff** is the difference of *"Pos"* and *"Neg"*. **BLUE** and **Pruple** indicate best and second-best performer respectively.

Models	QQP			WIKI.			MPRC		
	n=1	n=2	n=3	n=1	n=2	n=3	n=1	n=2	n=3
SBERT	0.898	0.831	0.775	0.945	0.909	0.874	0.929	0.879	0.829
USE	0.814	0.736	0.672	0.865	0.821	0.78	0.864	0.819	0.774
Infer-Sent	0.347	0.331	0.32	0.359	0.349	0.34	0.361	0.353	0.346
LASER	0.417	0.399	0.387	0.432	0.425	0.418	0.43	0.423	0.415
D2V	0.506	0.434	0.391	0.569	0.517	0.496	0.588	0.497	0.432
Bloom	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006
GPTNeo	0.273	0.266	0.259	0.277	0.272	0.267	0.278	0.274	0.269
GPT3 Ada	0.894	0.869	0.851	0.915	0.904	0.894	0.916	0.905	0.895
LlaMa-2	0.443	0.393	0.347	0.462	0.433	0.398	0.463	0.43	0.388

Table 3: Criterion 2: Normalized 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 **purple** indicate the best and second-best performer.

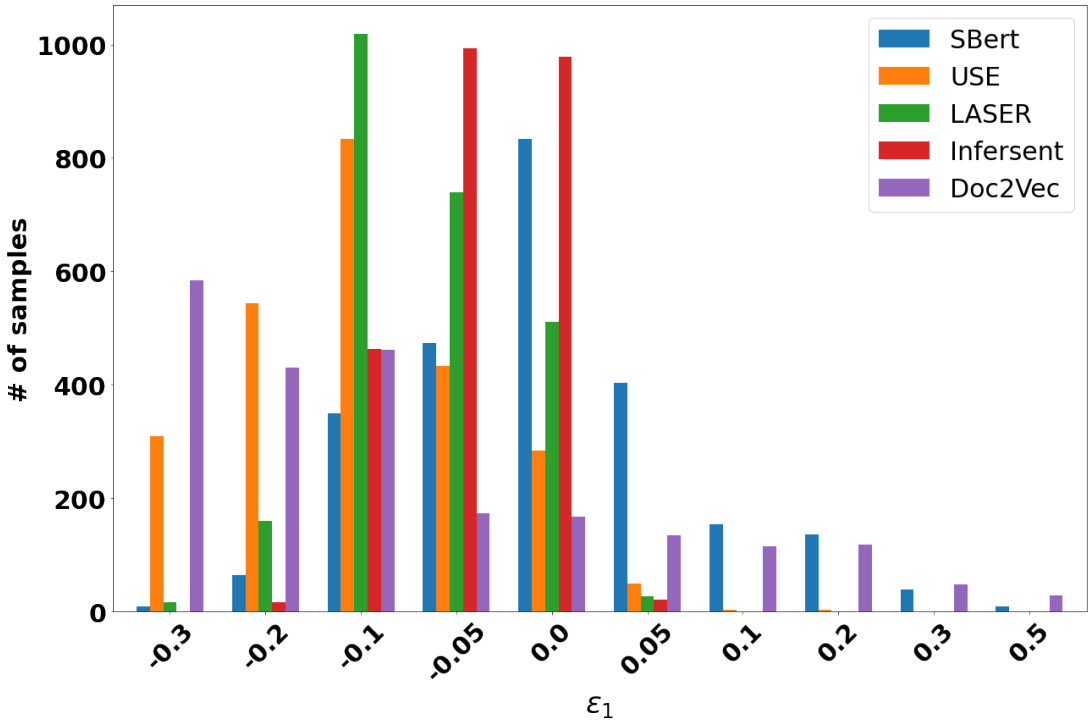
planation. On the whole, our findings suggest that while emergent models have achieved incremental advancements over classical models in Criterion-3, substantial opportunities for refinement remain.

A.4.4 Criteria 4

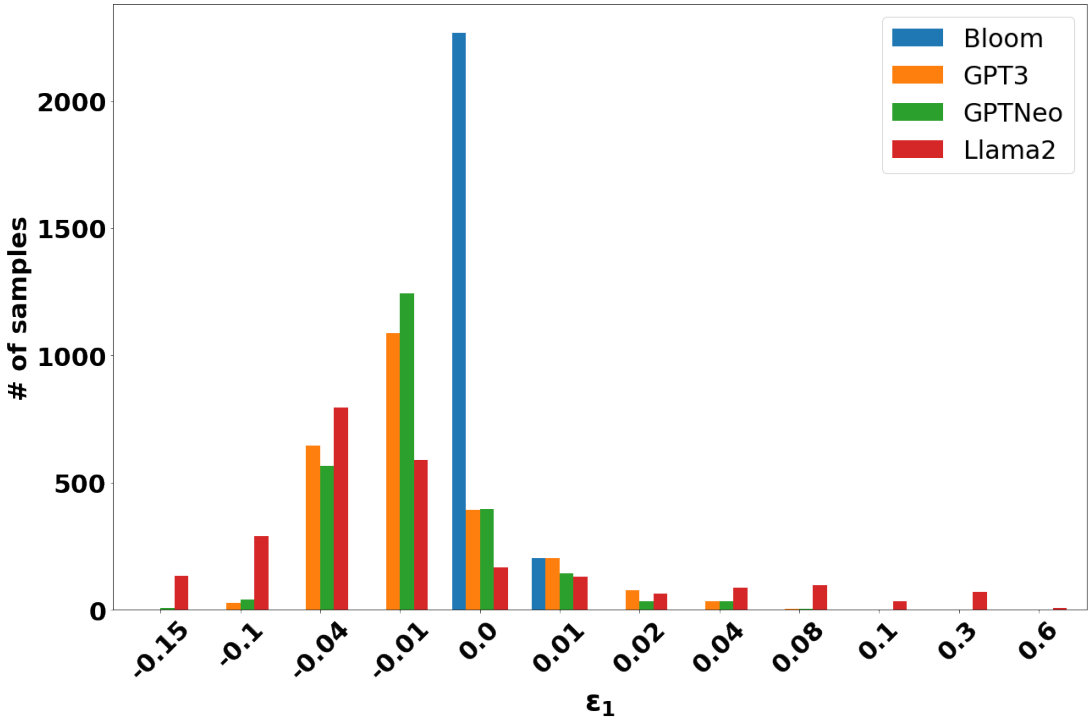
The Afin dataset, with its negation-affirmation sentence pairs, is an interesting criterion for evaluating encoder representations of semantic equivalence with lexical changes. Despite one sentence having negation, we anticipate high similarity between the pairs as they are paraphrases. The normalized cosine similarity scores in Table 4 show SBERT outperforming all models, followed by USE (0.799 vs. 0.695). Classical encoders like LASER and InferSent surpass emerging models, highlighting challenges for large language models in encoding sentence-level semantics.

A.4.5 Criteria 5

The results of the cosine similarity difference for the Jumble Sentence task are shown in Figure [5 - 12]. 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 $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_2$. 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 emergent models show some progress over classic models. The figures display the number of samples with a difference in cosine similarity score greater than ϵ_2.

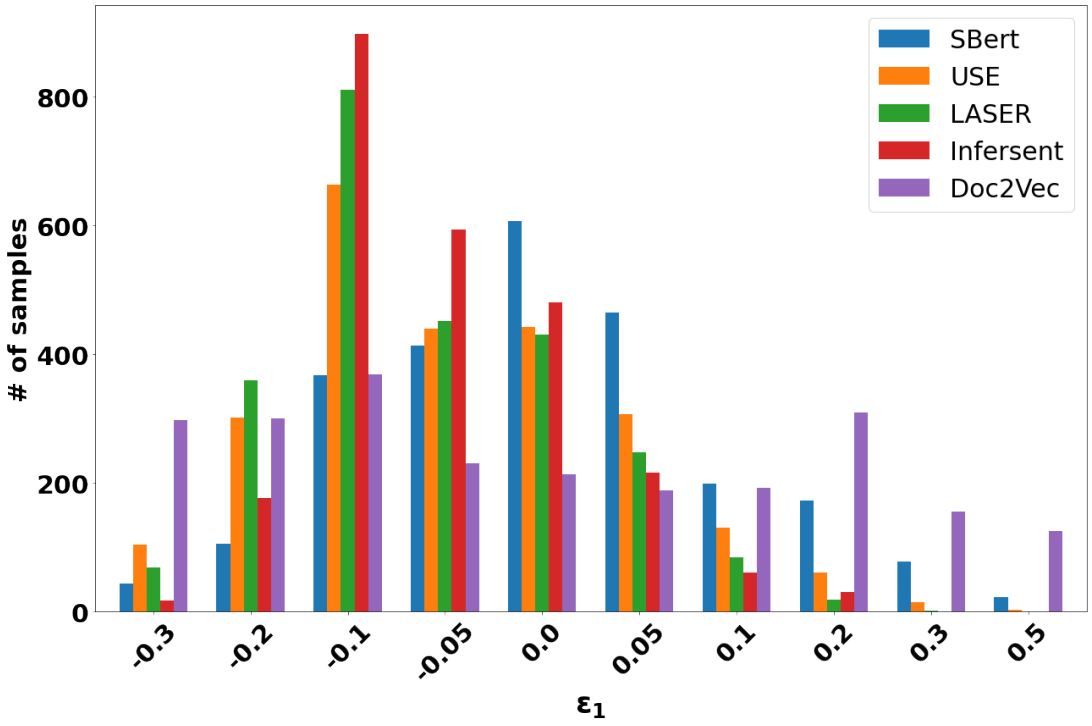


(a) Classical Model - Antonym Replacement Task on MRPC dataset

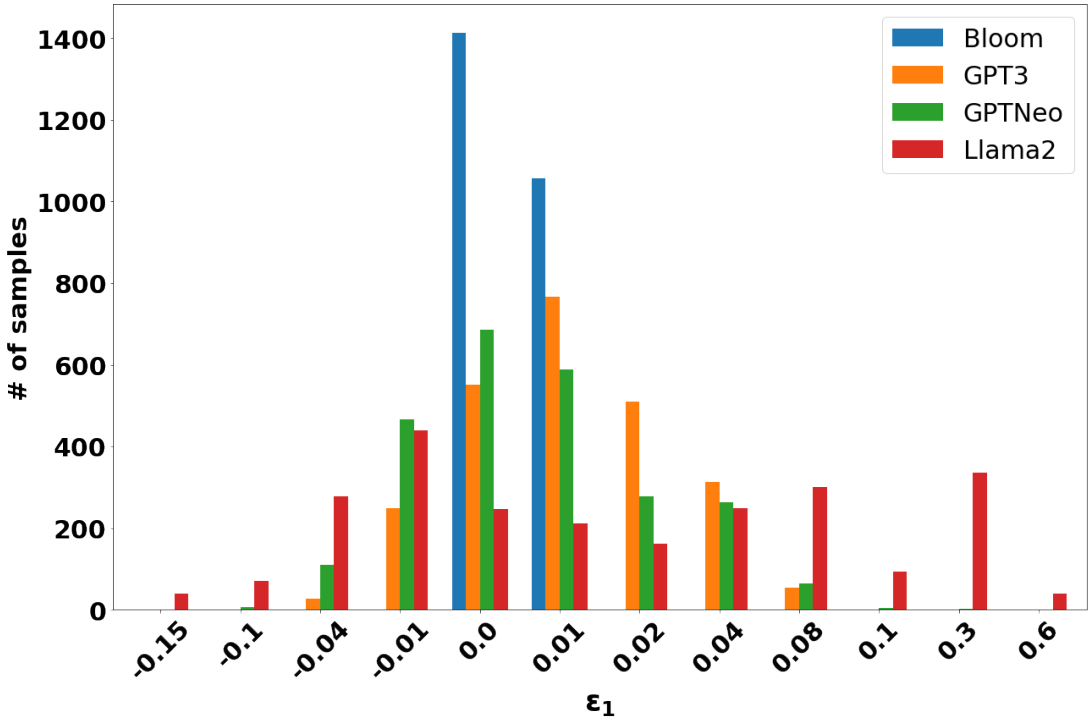


(b) Emergent Model - Antonym Replacement Task on MRPC dataset

Figure 2: Histogram plots for the Antonym Replacement Criterion-3 on MRPC dataset. (a) Classical Encoders and (b) Emergent Encoders. It highlights their ability to distinguish between a sentence and its antonym counterpart on MRPC. The scores are computed using the formula $Sim(S, S'_P) - Sim(S, S'_A) > \epsilon_1$, where ϵ_1 denotes the expected minimum margin of differentiation.



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

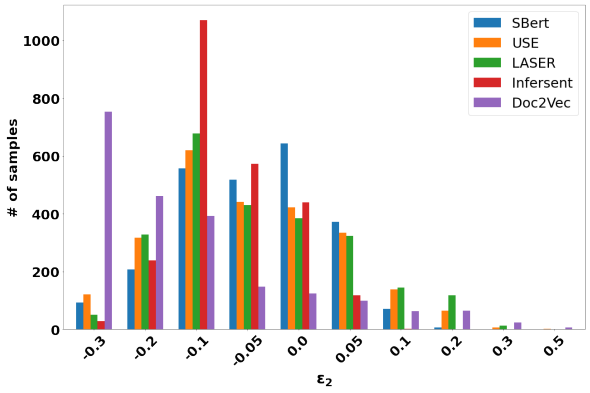


(b) Emergent Model - Antonym Replacement Task on PAWS-WIKI dataset

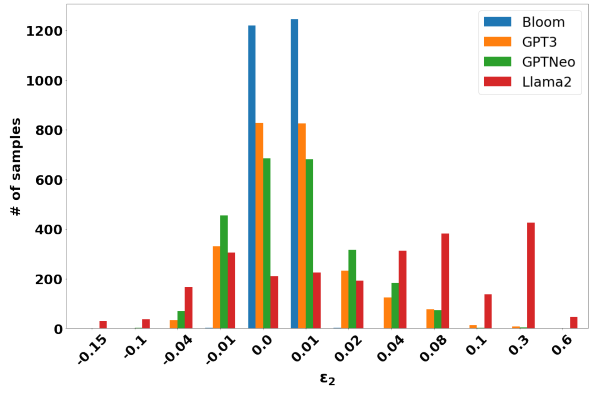
Figure 3: Histogram plots for the Antonym Replacement Criterion-3 on PAWS-WIKI dataset. (a) Classical Encoders and (b) Emergent Encoders. It highlights their ability to distinguish between a sentence and its antonym counterpart on PAWS-WIKI. The scores are computed using the formula $Sim(S, S'_P) - Sim(S, S'_A) > \epsilon_1$, where ϵ_1 denotes the expected minimum margin of differentiation.

Model	USE	SBERT	Infer-sent	LASER	D2V	Bloom	GPTNeo	GPT3-Ada	LlaMa2
Avg. Sim. score	**0.695**	**0.779**	0.325	0.387	-0.001	0.006	0.267	0.260	0.423

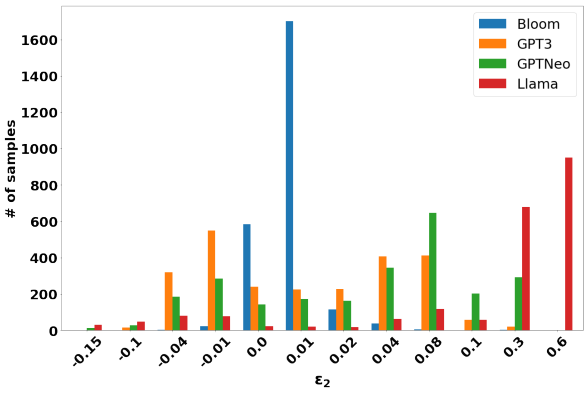
Table 4: Criterion-4: Normalized Avg. similarity score of negation-affirmative sentence pair sentences from the AFIN dataset. The **blue** and **purple** indicate the best and second-best performer.



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

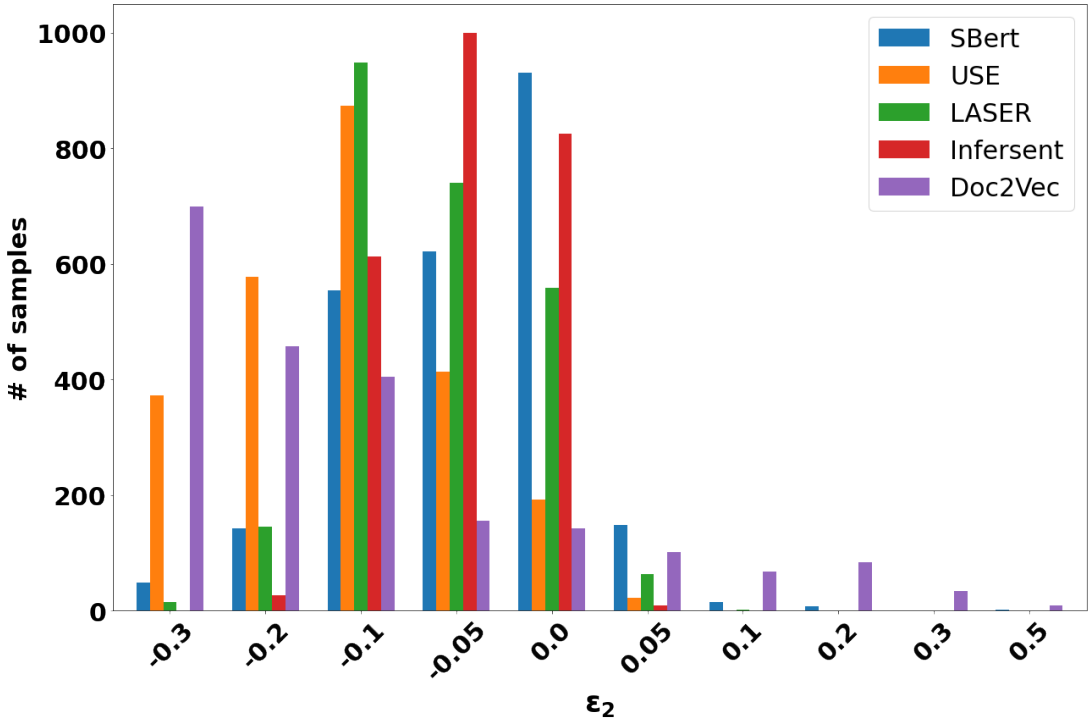


(b) Emergent Models- Sentence Jumb. on QQP for $n = 1$

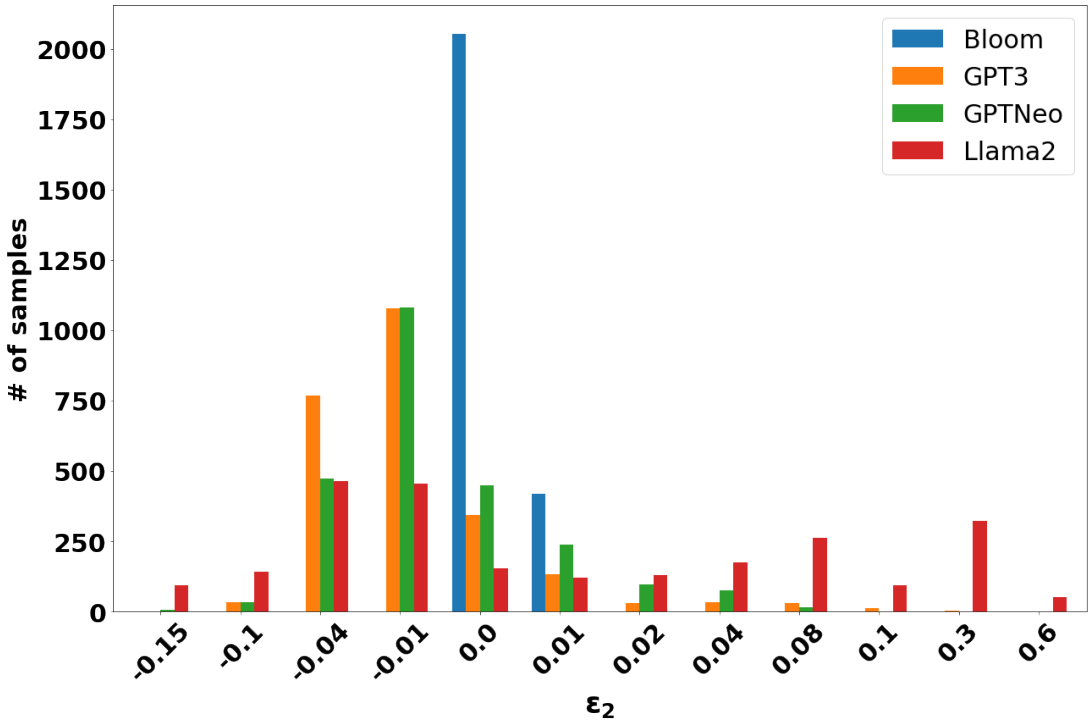


(c) Emergent Models- Sentence Jumb. on QQP for $n = 3$

Figure 4: The figures demonstrate the cosine similarity difference for Paraphrased Vs. Jumbling criterion. The score are calculated based on $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_2$. On the x-axis, the data is grouped into bins, and each bin represents the samples that fall within that ϵ_1. Appendix **??** presents the figure for the remaining QQP, MRPC, and PAWS-WIKI dataset

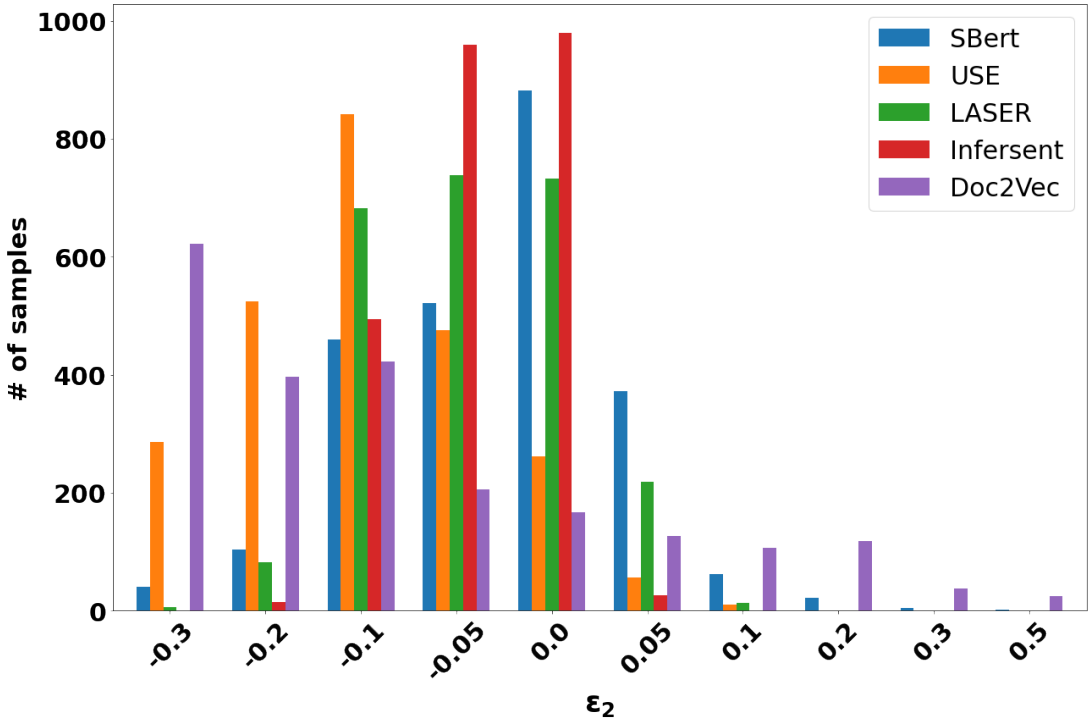


(a) Classical Model - Sentence Jumbling Task on MRPC dataset with n=1.

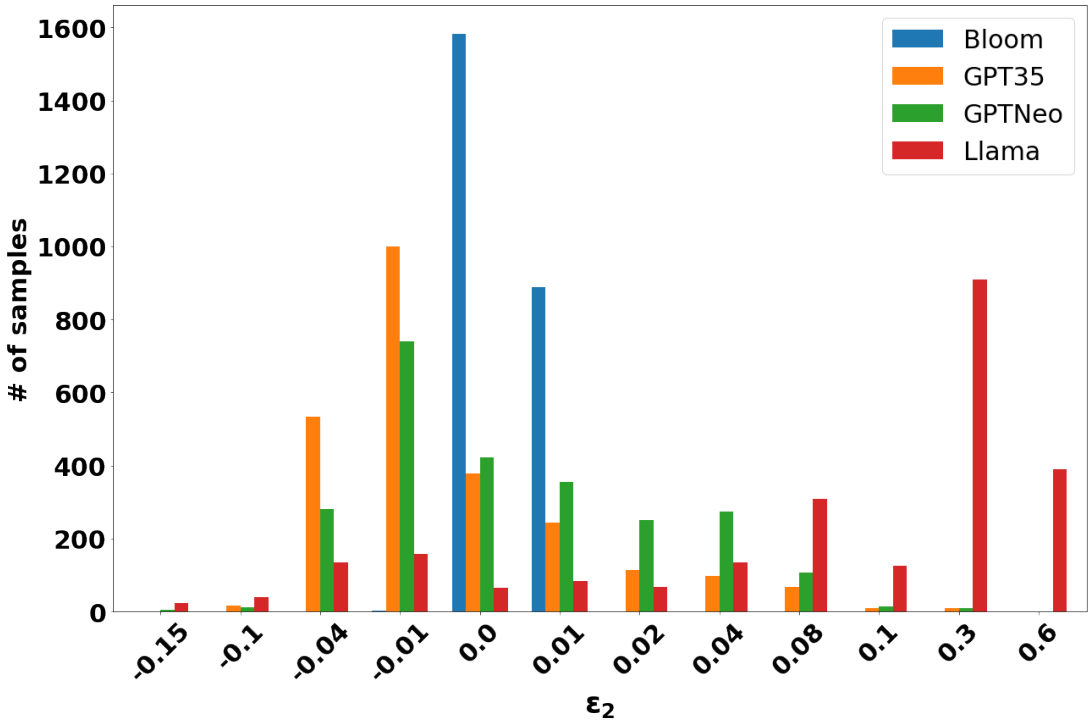


(b) Emergent Model - Sentence Jumbling Task on MRPC dataset with n=1.

Figure 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 emergent encoders, 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 $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_2$ 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.

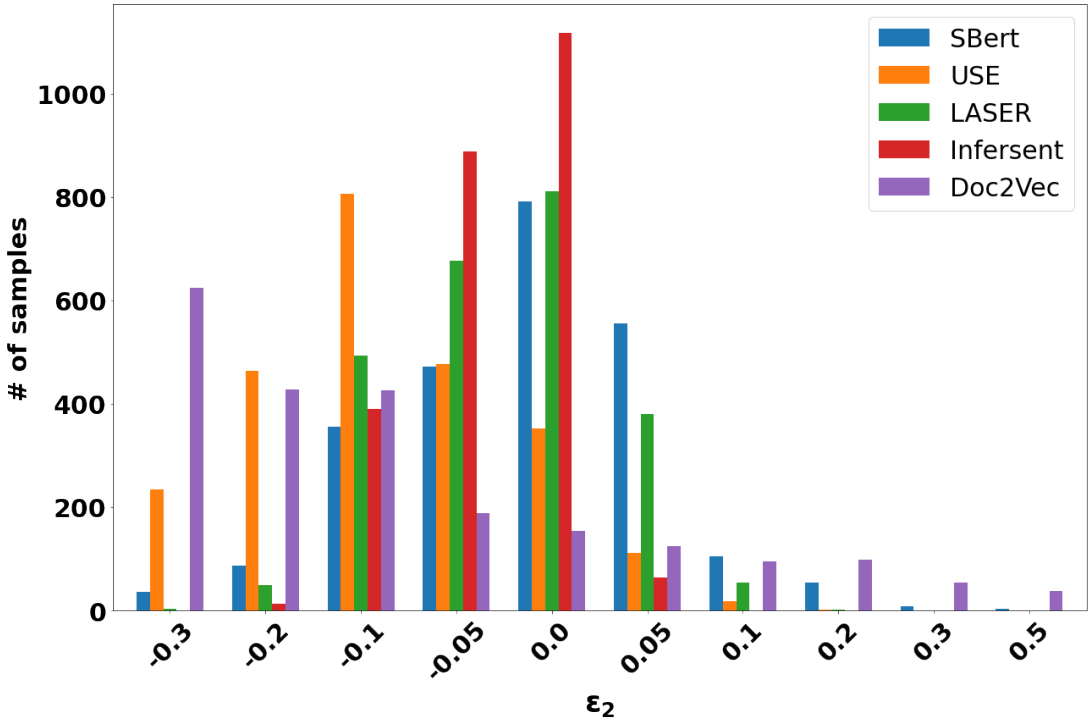


(a) Classical Model - Sentence Jumbling Task on MRPC dataset with n=2.

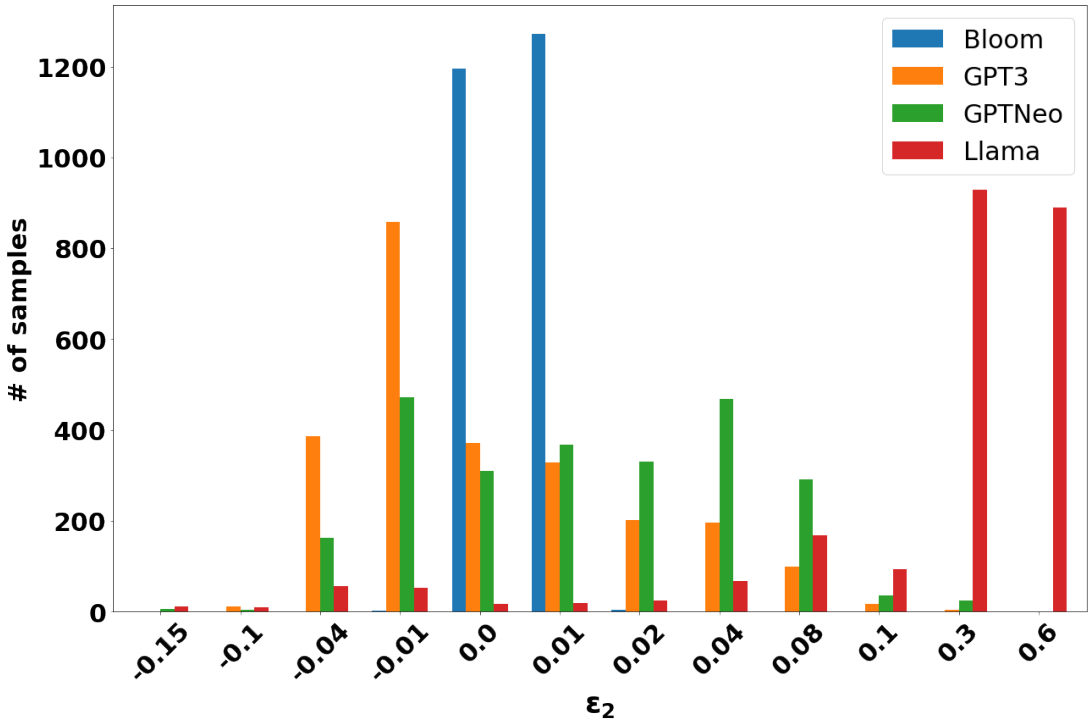


(b) Emergent Model - Sentence Jumbling Task on MRPC dataset with n=2.

Figure 6: 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 emergent encoders, 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 $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_2$ 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.

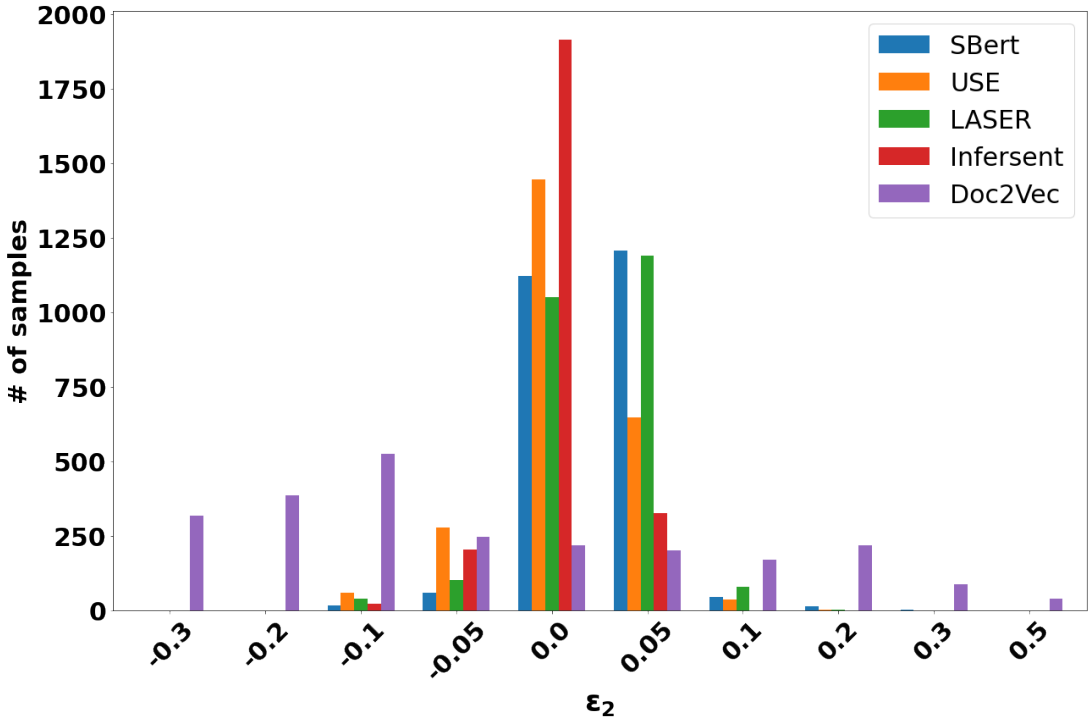


(a) Classical Model - Sentence Jumbling Task on MRPC dataset with n=3.

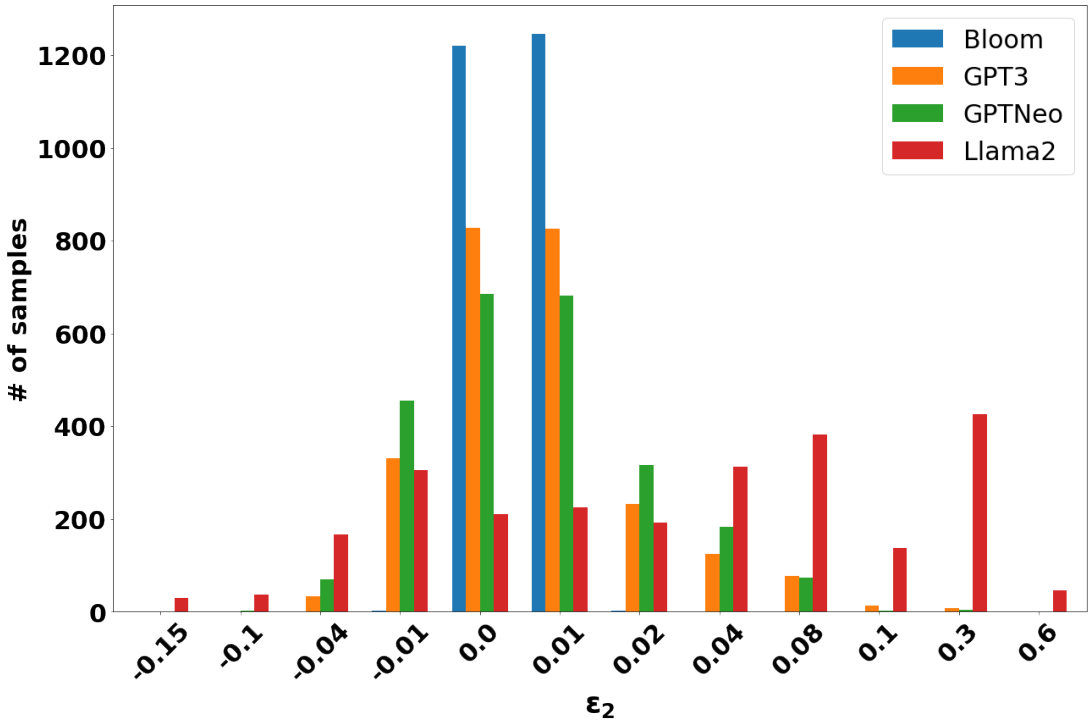


(b) Emergent Model - Sentence Jumbling Task on MRPC dataset with n=3.

Figure 7: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for MRPC dataset. Figures (a) and (b) depict histograms for classical and emergent encoders, 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 $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_2$ 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.

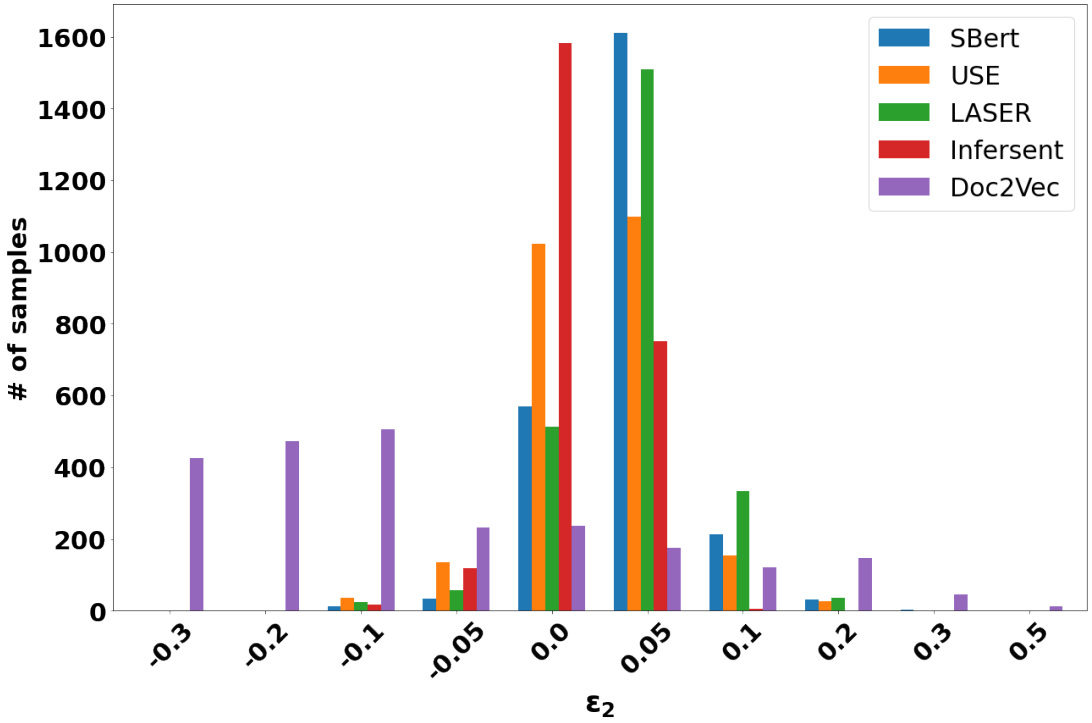


(a) Classical Model - Sentence Jumbling Task on PAWS-WIKI dataset with n=1.

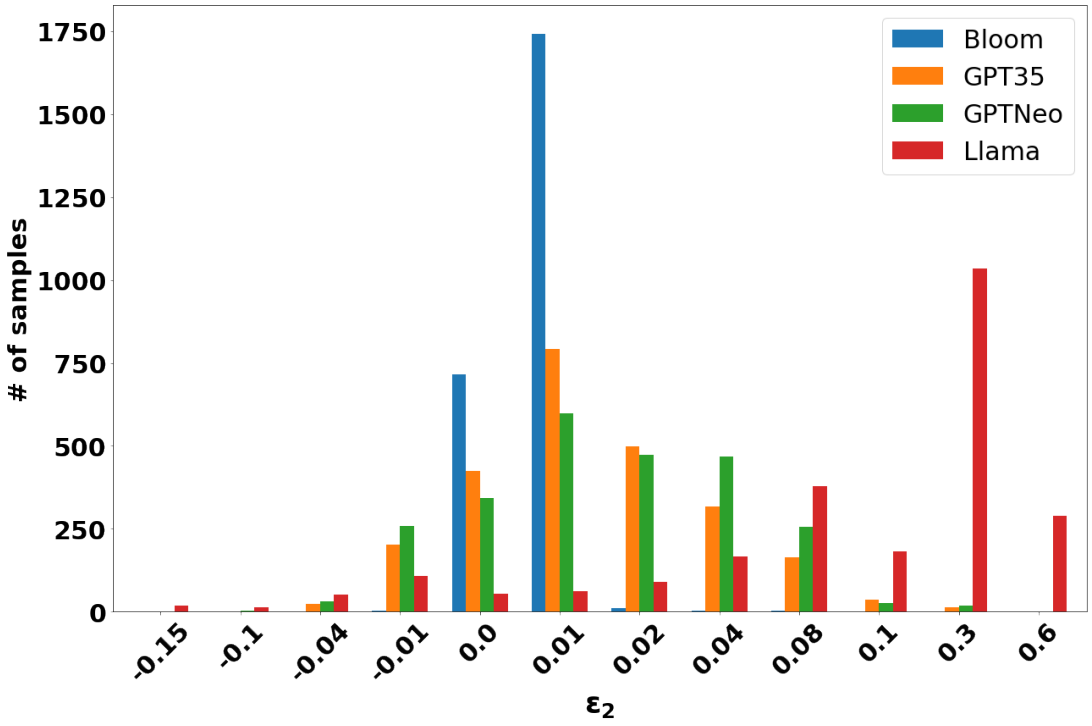


(b) Emergent Model - Sentence Jumbling Task on PAWS-WIKI dataset with n=1.

Figure 8: 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 emergent encoders, 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 $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_2$ 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.

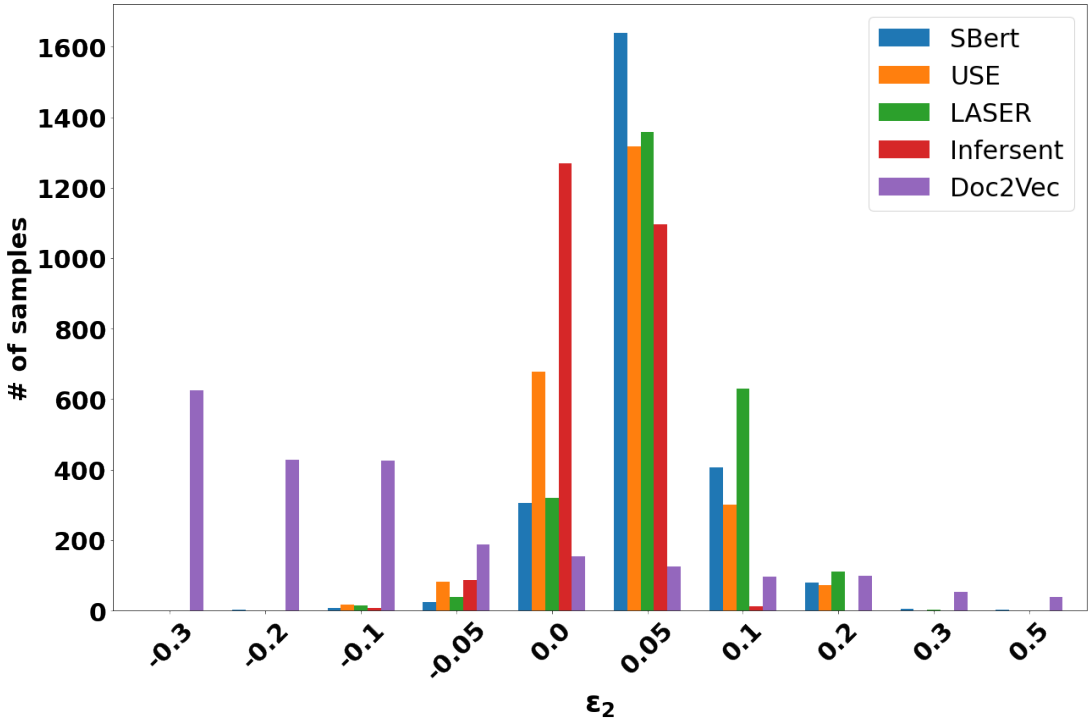


(a) Classical Model - Sentence Jumbling Task on PAWS-WIKI dataset with n=2.

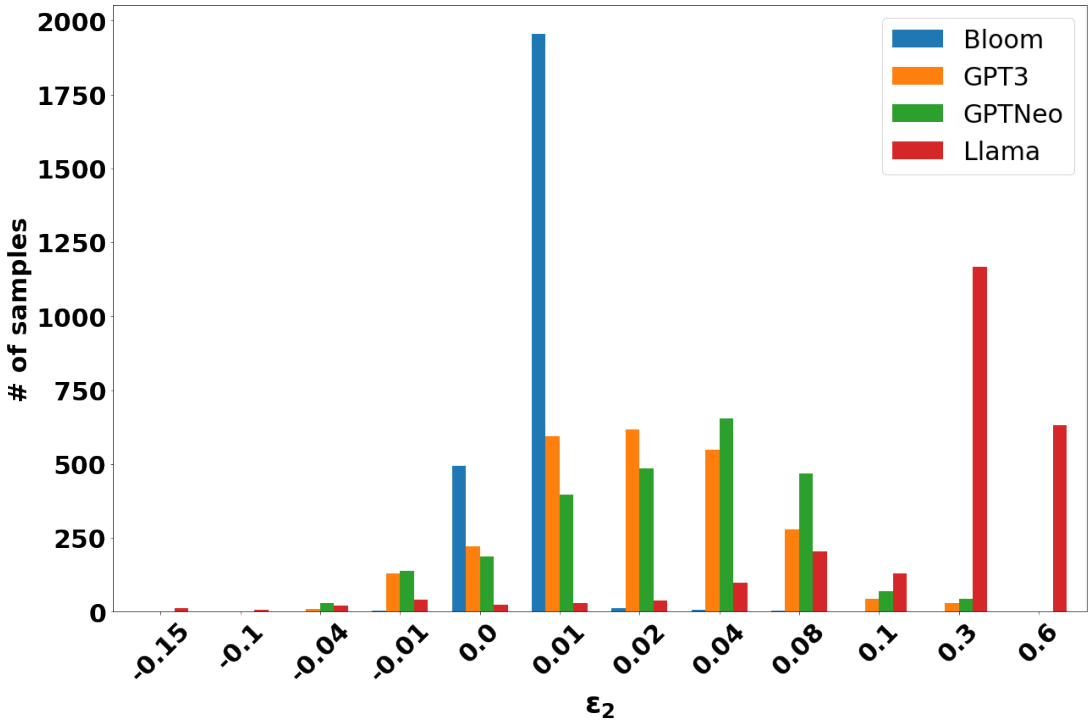


(b) Emergent Model - Sentence Jumbling Task on PAWS-WIKI dataset with n=2.

Figure 9: 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 emergent encoders, 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 $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_2$ 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.

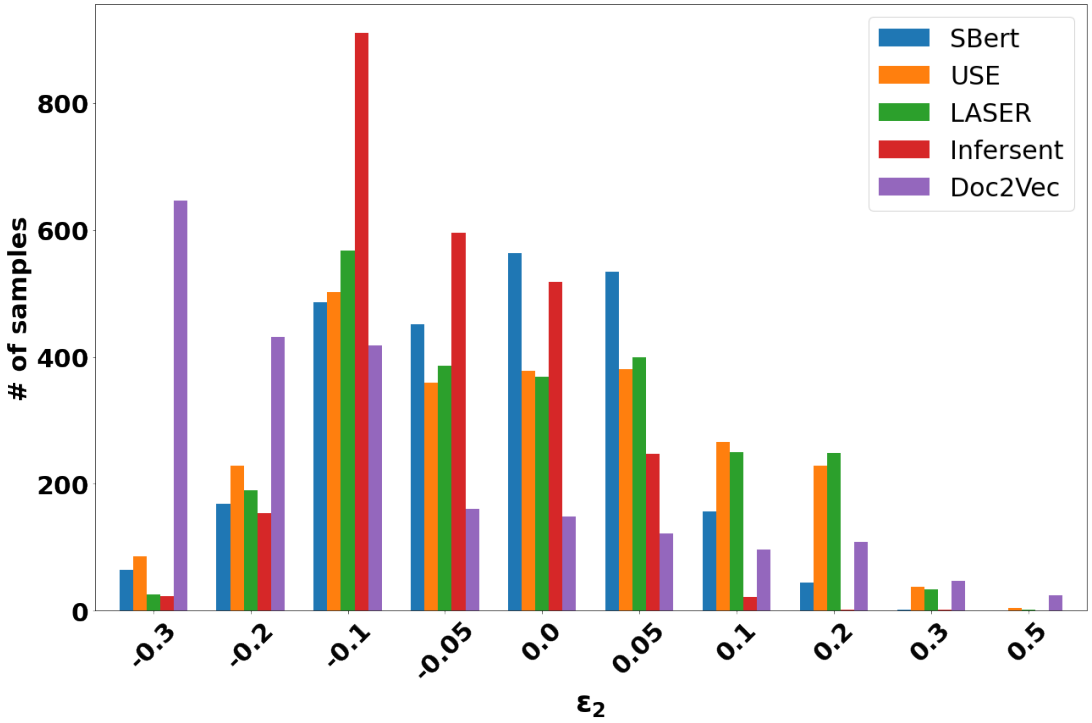


(a) Classical Model - Sentence Jumbling Task on PAWS-WIKI dataset with n=3.

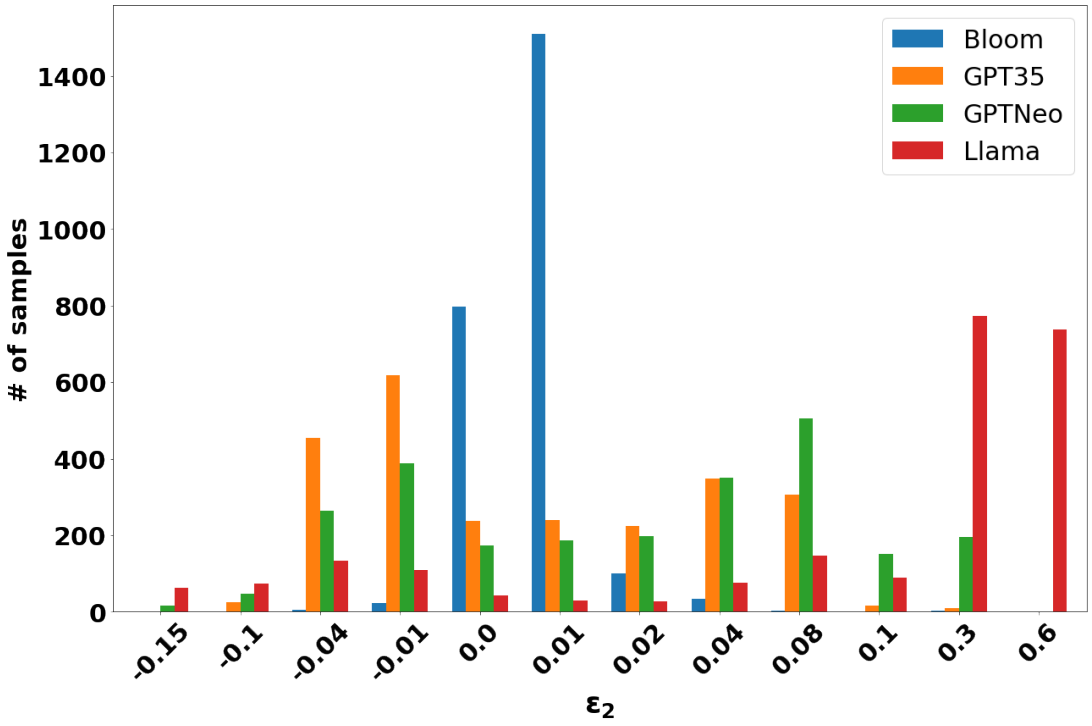


(b) Emergent Model - Sentence Jumbling Task on PAWS-WIKI dataset with n=3.

Figure 10: 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 emergent encoders, 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 $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_2$ 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.

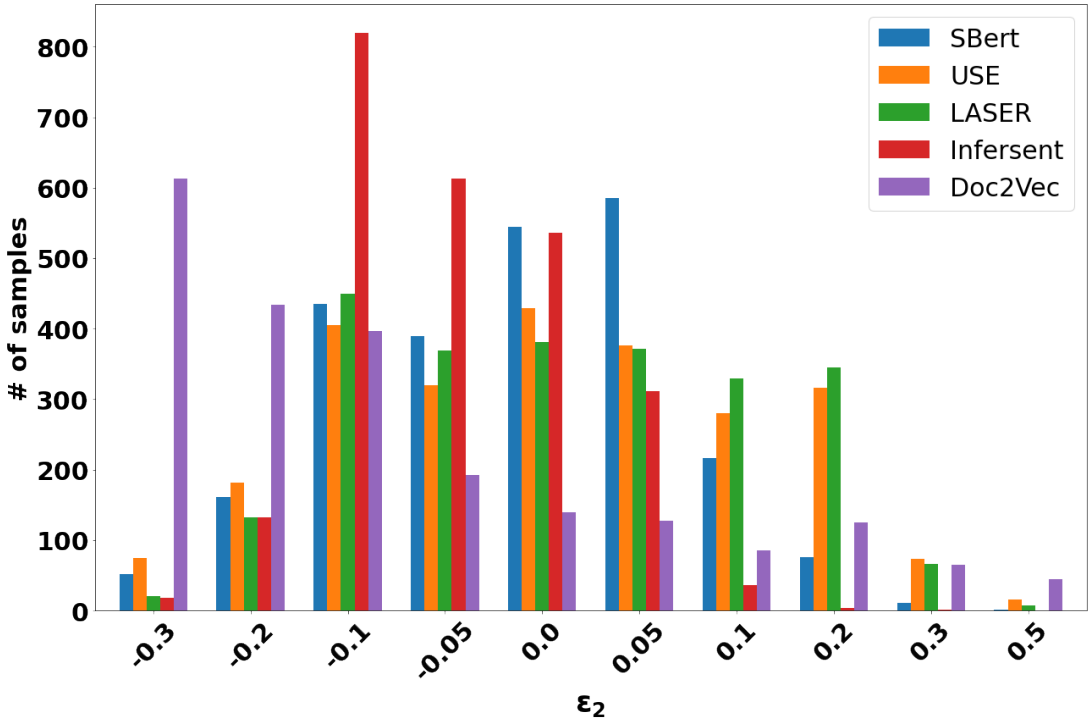


(a) Classical Model - Sentence Jumbling Task on QQP dataset with n=2.

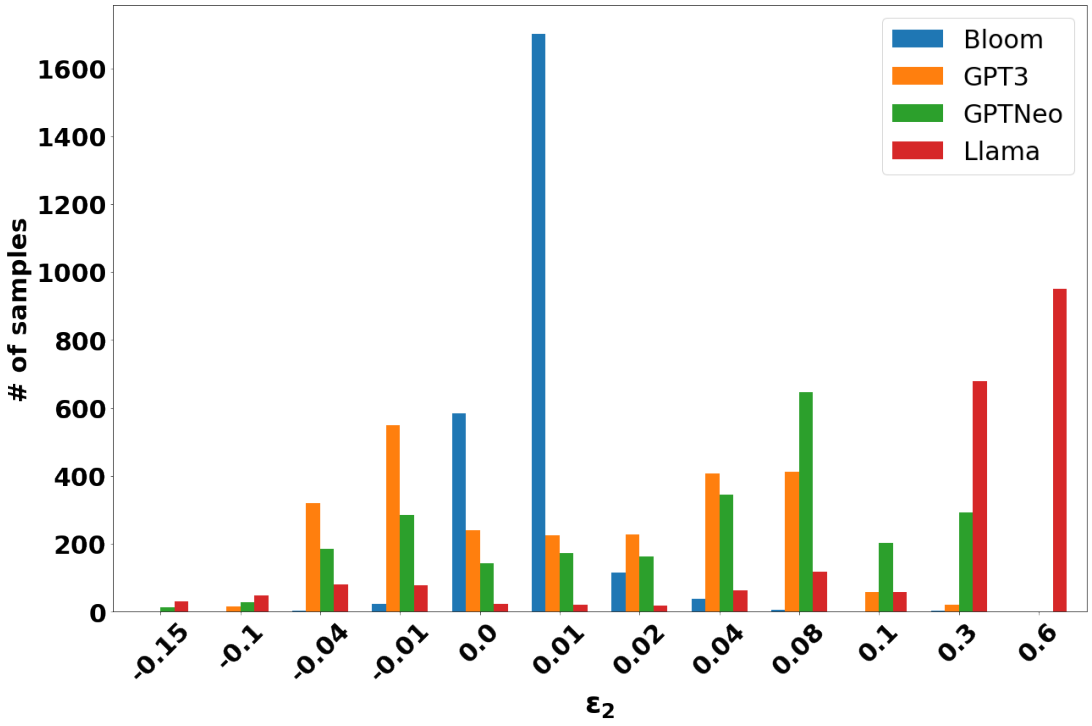


(b) Emergent Model - Sentence Jumbling Task on QQP dataset with n=2.

Figure 11: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for QQP dataset. Figures (a) and (b) depict histograms for classical and emergent encoders, 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 $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_2$ 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) Classical Model - Sentence Jumbling Task on QQP dataset with n=3.

Figure 12: The presented figures illustrate the results for the Jumble Sentence Criterion-5 for QQP dataset. Figures (a) and (b) depict histograms for classical and emergent encoders, 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 $Sim(S, S'_P) - Sim(S, S'_J) > \epsilon_2$ 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.