# Reweighting Strategy based on Synthetic Data Identification for Sentence Similarity Comparison

Anonymous ACL submission

## Abstract

Semantically meaningful sentence embeddings are important for numerous tasks in natural language processing. To obtain such embeddings, recent studies explored the idea of utilizing synthetically generated data from pretrained language models (PLMs) as a training corpus. However, PLMs often generate unrealistic sentences (*i.e.*, sentences different from human-written sentences). We hypothesize that training a model with these unrealistic sentences can have an adverse effect on learning semantically meaningful embeddings. To analyze this, we first train a classification model that identifies unrealistic sentences and observe that the linguistic features of the sentences predicted as unrealistic are significantly different from those of human-written sentences. Based on this, we propose a novel approach that first trains the classifier to measure the importance of each sentence. The distilled information from the classifier is then used to train a reliable sentence embedding model. Through extensive evaluation on four real-world datasets, we demonstrate that our model trained on synthetic data generalizes well and outperforms the baselines.

## 1 Introduction

High-quality sentence embeddings are essential to diverse applications in natural language processing (Cer et al., 2018; Reimers and Gurevych, 2019). However, obtaining a large amount of human-annotated datasets to train a sentence embedding model is difficult and expensive. To address this, Schick and Schütze (2021) recently introduced a method to train a sentence embedding model on synthetic data generated from pretrained language models (PLMs). However, PLMs sometimes produce unrealistic sentences different from human-written ones (Solaiman et al., 2019; Holtzman et al., 2019; Fagni et al., 2020). Therefore, training a model on synthetic data from PLMs may lead to performance degradation in various natural language processing tasks, but the study on the impact of such unrealistic data remains under-explored.

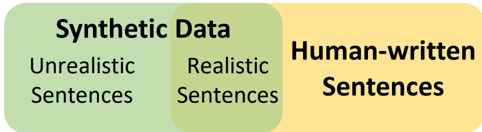


Figure 1: Sentences generated from the PLMs can be either realistic or unrealistic. Unrealistic sentences are distinct from human-written ones, whereas realistic sentences can be considered a subset of human-written sentences. We explore the effect of reducing the adverse effects of unrealistic sentences when training a model.

To this end, we first provide an in-depth analysis to demonstrate the shift of synthetic samples (both realistic and unrealistic) from the human-written sentences. In particular, we train a classifier (*i.e.*, Synthetic Data Identification (SDI) model) that identifies synthetic data from human-written sentences and observes that the linguistic features of the sentences predicted as unrealistic are much different from the human-written sentences compared to the linguistic features of the sentences predicted as realistic. Figure 1 presents an illustration to demonstrate different sentence distributions. Based on this analysis, we propose a simple method, **R**eweighting Loss based on **I**mportance of Machine-written **SE**ntence (RISE), which first utilizes the trained SDI model to measure the importance of each sentence in learning semantically meaningful sentence embeddings. Then, we utilize this distilled information from the SDI model and propose a data-item-level reweighting strategy to train a reliable sentence embedding model.

We evaluate the performance of our method on four different sentence similarity comparison datasets. Extensive experiments show that our model outperforms baseline models and generalizes better than the baselines across all datasets.

To sum up, our contributions include:

- We analyze the linguistic features of machine-

| | STSb | | | QQP | | | MRPC | | |
|---|---|---|---|---|---|---|---|---|---|
| | $x_h$ | $p_D(x_m)\uparrow$ | $p_D(x_m)\downarrow$ | $x_h$ | $p_D(x_m)\uparrow$ | $p_D(x_m)\downarrow$ | $x_h$ | $p_D(x_m)\uparrow$ | $p_D(x_m)\downarrow$ |
| BLEU-N | 34.80 | 25.75 | 2.93 | 30.3 | 34.95 | 7.86 | 48.53 | 46.97 | 5.59 |
| Jaccard | 41.98 | 33.97 | 5.98 | 39.91 | 42.49 | 11.31 | 53.55 | 53.33 | 10.52 |
| Distinct-N | 44.53 | 35.93 | 17.03 | 38.10 | 25.23 | 24.10 | 44.63 | 32.10 | 22.00 |
| Zipf coeff. | 1.03 | 1.07 | 1.23 | 1.11 | 1.06 | 1.12 | 0.98 | 1.02 | 1.23 |

Table 1: Results for comparing the sentences in different group. Jaccard indicates Jaccard similarity score. The score of generated sentences far from human scores is highlighted in underline. BLEU-N and Distinct-N indicate the average score with different $N$. The full results are available in Appendix E.

written sentences (both unrealistic and realistic) compared to human-written sentences.

- We also propose a simple yet effective method that first utilizes the Synthetic Data Identification (SDI) model to measure the importance of machine-written sentences for learning semantically meaningful embeddings.
- We then propose a new loss term based on the importance of sentences to train a reliable sentence embedding model.
- We extensively evaluate our model on diverse datasets and observe that our method consistently enhance sentence encoder performance trained on synthetic datasets.

## 2 Related Work

Synthetic data generation using pretrained language models has shown promising results in various natural language processing tasks (Yang et al., 2020; Papanikolaou and Pierleoni, 2020; Ding et al., 2020; Edwards et al., 2021; Chang et al., 2021). Recently, Schick and Schütze (2021) proposed a new method, DINO, to generate a synthetic dataset for textual semantic similarity task. Another recent work, Yoo et al. (2021) proposed a new data augmentation framework for sentence classification by leveraging a large-scale PLM (Brown et al., 2020). However, synthetic data can be misused in malicious usage, such as fake news generation. To prevent such a fraudulent use, recent studies (Zellers et al., 2019; Weiss, 2019; Uchendu et al., 2020; Adelani et al., 2020) aim to detect the synthetically generated text. On the contrary, we identify unrealistic sentences from machine-written data and mitigate their influence to achieve accurate and robust learning. While Yi et al. (2021) suggested assigning high weights to challenging examples in a data augmentation setup, our work mainly focuses on using only synthetic samples from PLMs.

## 3 Analysis on Synthetic Sentences

This section describes the generation of the synthetic dataset, followed by training the model to identify synthetic sentences from human-written ones. Then, we present a novel analysis to demonstrate the shift of synthetic samples (both realistic and unrealistic) from the human-written sentences.

**Synthetic Data Generation.** To obtain machine-generated sentences, we leverage the ability of prompt-based zero-shot generation in a generative PLM (Radford et al., 2019) (Figure 2 A). Specifically, given a sentence $x_h \in C_{src}$ where $C_{src}$ is a set of human-written sentences and the target similarity level $y \in Y$, this framework produces a sentence $x_m \in X_m$ that has semantic similarity with $x_h$ equal to the target similarity level $y$. The generated examples $\{x_m, x_h, y\}$ are later used to train a model for sentence similarity comparison tasks.

For generating a synthetic dataset, we use Semantic Textual Similarity benchmark (STSb) (Cer et al., 2017), Quora Question Pairs (QQP) [1], and Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005) to obtain a corpus of human-written sentences. We follow the details for the data generation process in Schick and Schütze (2021). Through this synthetic data generation process, we obtain about 76k, 78k, and 55k examples of STSb, QQP, and MRPC datasets, respectively.

**Synthetic Data Identification (SDI).** We now train a binary classification model $D$ based on a bi-directional PLM (Devlin et al., 2019) to distinguish machine-written sentences from human-written sentences (Figure 2 B). We refer to this model as the Synthetic Data Identification (SDI) model and train it separately for each $C_{src}$. We use machine-written sentences $X_m$ and human-written

[1] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

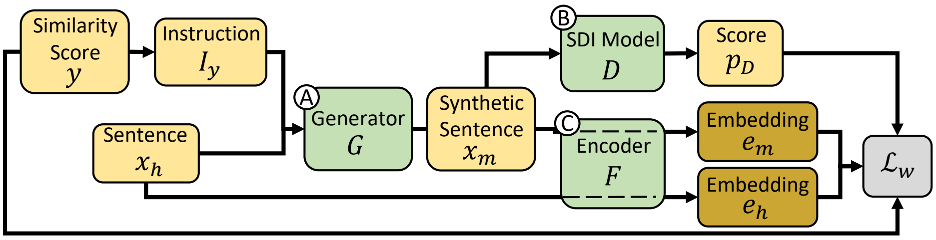


Figure 2: Overview of **RISE**. We feed an instruction $I_y$ and a human-written sentence $x_h$ to the Generator $G$ which produces a machine-written sentence $x_s$. We then measure importance score $p_D$ using $x_s$ as input. Finally, we predict the similarity score using the embedding vector of $x_s$ and $x_h$. We compute the loss and multiply $p_h$.

sentences $X_h$ in the same proportion for training the model.[2] We use the prediction confidence $p_D$ of the sentence to measure the degree of the shift of the generated sentences from the human-written sentences.

**Analysis.** We now analyze to demonstrate the shift of synthetic samples from the human-written sentences. We use the following metrics to analyze the lexical-level linguistic patterns of each sentence: (1) **BLEU** (Papineni et al., 2002) and **Jaccard Similarity** (Montahaei et al., 2019) that calculate the lexical-level similarity between $x_m$ and its paired sentence. (2) **Distinct-N** (Li et al., 2015) that calculates the ratio of unique N-grams among the total number of N-grams in each group for $x_m$. (3) **Zipf coefficient** (Holtzman et al., 2019) that calculates the Zipf coefficient to analyze the vocabulary usage for $x_m$. We utilize the prediction confidence $p_D$ from the SDI model to measure the importance of generated sentences in learning meaningful sentence embeddings. We select the top 10% ($p_D(x_m)$ ↑) and bottom 10% ($p_D(x_m)$ ↓) of the machine-written sentences based on their sorted importance and analyze their linguistic features.

Table 1 presents results to demonstrate that the unrealistic samples are significantly shifted from the human-written sentences. Further, we observe that except for Zipf coefficient in QQP dataset, generated sentences with high $p_D(x_m)$ always have scores close to the scores of human-written sentences ($x_h$) compared to the sentences with low $p_D(x_m)$. We provide a detailed analysis in Appendix E. Based on these observations, we confirm that there exist a large variance in terms of how much the sentences are shifted from human-written sentences. Therefore, it is critical to handle the generated sentences carefully so that the model is not biased to the sentences that are sufficiently different from human-written sentences (*i.e.*, unrealistic samples).

[2] The accuracy of classifiers of each dataset on the validation set are 77.87, 83.21, and 93.05% in STSb, MRPC, and QQP datasets, respectively.

## 4 Method

We now introduce a simple yet effective method, **R**eweighting Loss based on **I**mportance of Machine-written **SE**ntence (RISE), that aims to give less importance to unrealistic machine-written sentences than realistic sentences. Our method consists of two stages: (1) measuring the importance of the generated sentences in learning semantically meaningful embeddings using the prediction confidence $p_D$ from the SDI model (defined in Section 3); 2) utilizing the importance score to control the weight of the loss for each example during training so that the model does not deviate significantly from the distribution of the human-written text. The training procedure except for loss is the same as usual training of a sentence embedding model based on the bi-encoder architecture (Reimers and Gurevych, 2019). More details on training the sentence encoder are provided in Appendix C.

**Reweighting Loss using Importance Score.** We utilize the prediction confidence $p_D$ from the SDI model (Section 3) to measure the importance of generated sentences. In particular, we modify the loss to make the realistic machine-written examples (*i.e.*, examples with high scores) have more contribution to the loss, whereas the unrealistic machine-written examples (*i.e.*, examples with low score) have less contribution (in Figure 2 C). The loss of each example is defined as:

$$\mathcal{L}_{\mathrm{w}}(\theta_f) = p_D * \mathcal{L}(\theta_f), \quad (1)$$

where $\mathcal{L}(\theta_f)$ denotes the original loss of the sentence encoder $F$ for sentence similarity task, and $\mathcal{L}_w(\theta_f)$ denotes the modified loss by RISE. $\theta_f$ denotes the parameters of the sentence encoder. This

| $C_{src}$ | Model | STSb $r$ | STSb $\rho$ | QQP Acc. | QQP F1 | MRPC Acc. | MRPC F1 | PAWS F1 |
|---|---|---|---|---|---|---|---|---|
| ***STSb*** | DINO | 78.45 | 77.71 | 73.14 | 68.04 | 70.44 | 81.16 | 47.30 |
| | RISE | **79.11** (+0.66) | **78.57** (+0.86) | **74.47** (+1.33) | **69.08** (+1.04) | **72.84** (+2.4) | **82.01** (+0.85) | **50.24** (+2.94) |
| | ∟ Filtering | 77.73 (-0.72) | 77.45 (-0.26) | 73.06 (-0.08) | 67.94 (-0.10) | 68.96 (-1.48) | 81.35 (+0.19) | 46.72 (-0.58) |
| | ∟ Random | 79.03 (+0.58) | 78.39 (+0.68) | 73.09 (-0.05) | 68.03 (-0.01) | 71.09 (+0.65) | 81.62 (+0.46) | 50.17 (+2.87) |
| ***QQP*** | DINO | 64.93 | 65.93 | 73.20 | 67.72 | 70.75 | 80.40 | 44.47 |
| | RISE | **78.36** (+13.43) | **77.13** (+11.2) | 73.35 (+0.15) | 67.76 (+0.04) | **72.38** (+1.63) | **81.35** (+0.95) | 46.28 (+1.81 ) |
| | ∟ Filtering | 65.24 (+0.31) | 66.36 (+0.43) | **73.48** (+0.28) | **67.95** (+0.23) | 69.77 (-0.98) | 80.26 (-0.14) | 43.36 (-1.11) |
| | ∟ Random | 73.49 (+8.56) | 72.88 (+6.95) | 73.14 (-0.06) | 67.75 (+0.03) | 69.76 (-0.99) | 80.83( +0.43) | **46.97** (+2.5) |
| ***MRPC*** | DINO | 75.51 | 73.87 | 71.85 | 65.70 | 71.57 | 81.55 | 47.35 |
| | RISE | **77.47** (+1.96) | **76.86** (+2.99) | **74.23** (+2.38) | **68.82** (+3.12) | 71.97 (+0.4) | **81.95** (+0.4) | **49.35** (+2.00) |
| | ∟ Filtering | 76.25 (+0.74) | 74.88 (+1.01) | 71.05 (-0.80) | 64.82 (-0.88) | 71.34 (-0.23) | 80.76 (-0.79) | 47.84 (+0.49) |
| | ∟ Random | 76.06 (+0.55) | 74.51 (+0.64) | 72.52 (+0.67) | 66.45 (+0.75) | **72.19** (+0.62) | 81.71 (+0.16) | 47.56 (+0.21) |

Table 2: Evaluation results of different sentence embedding models on four sentence similarity task dataset. We highlight the best result within each $C_{src}$ in each metric as **bold**. The number in right bracket indicates the performance difference with DINO. For regression task, we use Pearson correlation ($r$) and Spearman's rank correlation coefficient ($\rho$) metrics are used for evaluation. Each score represents the average of five trials.

re-weighting procedure aims to adjust the influence of examples based on the extent of shift of the sentence from the human-written sentences.

## 5 Experimental Settings

We evaluate each model on Paraphrase Adversaries from Word Scrambling of Quora Question Pairs (Zhang et al., 2019) (PAWS-QQP) including STSb, QQP, and MRPC. It aims to evaluate the robustness of the model against adversarial attacks for the sentence similarity comparison task. We provide more details in the Appendix B.

We train a model to solve the sentence similarity task as a regression problem. However, since all datasets except STSb only contain discrete labels, we set threshold using the validation dataset to make a binary decision.

We apply our method to DINO and denote it as RISE. In addition to experiments with RISE, we conduct experiments with the following variants: (1) *Filtering*: We filter out the bottom 10% of the machine-written sentences based on their sorted importance. We then use the remaining examples for training without using our modified loss. (2) *Random*: We randomly sample a scalar value from $U(0, 1)$ for each example and use it as it's importance. DINO and variants of our method are based on sentence-RoBERTa-base architecture, and are fine-tuned on synthetic datasets only.

## 6 Results

Table 2 report the performance of our method and the baselines on the sentence similarity task. We observe that our model outperforms the strong baselines and improves the performance of models trained on synthetic datasets. These results support our assumption that reweighting the loss of each machine-written sentence based on it's importance would enhance the reliability of the model and making it less biased to unrealistic machine-written sentences. Especially, we find that the magnitude of improvement is usually higher when the model is evaluated on the human-annotated dataset from different domain than the source of training data $C_{src}$. These results imply that our method can enhance the robustness of the sentence encoder trained on a synthetic dataset when evaluated on dataset from different domain. In terms of the variants of our method, using the randomly sampled scalar value as an importance score usually degrades performance. In addition, models that filter out unrealistic examples and train without using RISE shows lower performance than RISE. Based on these observations, we confirm that training the model using RISE enhances the reliability of the model.

## 7 Conclusion

In this paper, we confirm that the linguistic features of unrealistic machine-written sentences are dissimilar to those of human-written sentences. Based on this, we propose a new method to reweight the loss based on the importance of the sentences from synthetic data identification (SDI) model for learning semantically meaningful embeddings. The extensive experiments show that RISE achieves performance gains over strong baselines, and the results show the robustness of our model.

## References


David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection. In *International Conference on Advanced Information Networking and Applications*, pages 1341–1354. Springer.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

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

Ernie Chang, Xiaoyu Shen, Dawei Zhu, Vera Demberg, and Hui Su. 2021. Neural data-to-text generation with lm-based text augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 758–768.

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.

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.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. DAGA: Data augmentation with a generation approach for low-resource tagging tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6045–6057, Online. 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)*.

Aleksandra Edwards, Asahi Ushio, Jose Camacho-Collados, Hélène de Ribaupierre, and Alun Preece. 2021. Guiding generative language models for data augmentation in few-shot text classification. *arXiv preprint arXiv:2111.09064*.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2020. Tweepfake: about detecting deepfake tweets. *arXiv preprint arXiv:2008.00036*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. 2018. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pages 2805–2814. PMLR.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models.

Yannis Papanikolaou and Andrea Pierleoni. 2020. Dare: Data augmented relation extraction with gpt-2. *arXiv preprint arXiv:2004.13845*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Conf. on Empirical Methods in Natural Language Processing (EMNLP)*.

Max Weiss. 2019. Deepfake bot submissions to federal public comment websites cannot be distinguished from human submissions. *Technology Science*.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.

Mingyang Yi, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, and Zhi-Ming Ma. 2021. Reweighting augmented samples by minimizing the maximal expected loss. In *Proc. the International Conference on Learning Representations (ICLR)*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. GPT3Mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 9054–9065.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. 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 1298–1308.

## Appendix

## A Training Details

**Environment Details** All experiments in Table 2 in the main paper is implemented in Ubuntu 18.04.4 LTS, 3090 RTX GPU with 24GB of memory, and AMD EPYC 7702. The version of libraries we experiment are 3.8 for python and 1.4.0 for pytorch. We implemented all models with PyTorch using Sentence-Transformers[3] library from Ubiquitous Knowledge Processing Lab.

**Training and Evaluation.** We train a model to solve sentence similarity task as a regression problem. However, since all the datasets except STSb only contain discrete labels, we set the threshold using validation dataset to make binary decision. Training a model takes 5 minutes per epoch.

**Hyperparameter Details** The DINO are reproduced as described in the previous works. To compute sentence simiarity score, we use cosine similarity as distance metric. We search the best hyperparameters using grid search. During the prediction of SDI model, we use use the temperature scaling ($\tau$) (Kumar et al., 2018) is applied before softmax function. The best hyperparameters for each dataset of **RISE** are described as below:

| Hyperparameter | STSb | QQP | MRPC |
|---|---|---|---|
| batch size | 32 | 32 | 32 |
| learning rate | 2e-5 | 2e-5 | 2e-5 |
| number of epochs | 3 | 3 | 3 |
| temperature $\tau$ | 0.5 | 0.9 | 0.7 |

Table 3: Hyperparameters used in experiments. We conduct grid search to find the best hyperparameter settings.

## B Datasets Details

As aforementioned in Section 3, STSb (Cer et al., 2017), QQP, and MRPC (Dolan and Brockett, 2005) are used to obtain a corpus of human-written sentences. The size of corpus $|C_{src}|$ is equally set to 10,000 across datasets. The set of similarity level $Y$ is $\{0, 0.5, 1\}$. We generate samples from corpus

**Sentence Textual Simiarlity benchmark(STSb)** (Cer et al., 2018) consists of sentence pairs drawn from news, video and image captions, and natural language inference data. Each pair is human-annotated with a continuous score from 1 to 5; the task is to predict these scores. In this experiment, we normalize the original similarity score to have from 0 to 1. We evaluate using Pearson and Spearman correlation coefficients.

**Quora Question Pairs(QQP)** [4] consists of question pairs from the community Quora. The task is to classify that a pairs of question have semantically same meaning.

**Microsoft Research Paraphrase Corpus(MRPC)** (Dolan and Brockett, 2005) is a corpus of sentence pairs from online news sources, with human annotations for whether the sentences in the pair are semantically same. The class have the imbalanced distribution.(68% positive).

**Paraphrase Adversaries from Word Scrambling of Quora Question PAWS-QQP (Zhang et al., 2019)** contains human-labeled and noisily labeled pairs that feature the importance of modeling structure, context, and word order information for the problem of paraphrase identification. The dataset has two subsets, one based on Wikipedia and the other one based on the Quora Question Pairs (QQP) dataset. In this paper, we only use examples based on QQP. The class have the imbalanced distribution.(31.3% positive).

| Data | STSb | QQP | MRPC | PAWS-QQP |
|---|---|---|---|---|
| $X_m^{train}$ | 76.9k | 78.2k | 55.3k | - |
| $X_m^{dev}$ | 59.2k | 78.3k | 6.3k | - |
| $X_{src}^{dev}$ | 1.5k | 18.1k | 0.4k | 0.3k |
| $X_{src}^{test}$ | 1.4k | 40.4k | 1.7k | 0.3k |

Table 4: Dataset statistics. The class distribution of MRPC, QQP, and PAWS-QQP is imbalanced.

## C Training Sentence Encoder for Sentence Similarity Task

Sentence similarity task aims to determine the similarity between two sentences. It can be formulated by classifying whether the two sentences are semantically similar or not or by measuring the distance between two sentences. A common and scalable approach for this task is based on Bi-encoder architecture (Reimers and Gurevych, 2019) which involves converting the sentences into embedding vectors and then measuring the similarity between sentences by calculating the distance between them in the embedding space.

More formally, given two sentences $s_1$ and $s_2$, and their ground truth similarity score $y$, a sentence encoder $F$ encodes the sentences, $s_1$ and $s_2$, into

[3] https://github.com/UKPLab/sentence-transformers

[4] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

their embedding vectors, $e_1$ and $e_2$, respectively. A distance metric $d$ is then used to measure their similarity score $\hat{y}$, which is defined by:

$$\hat{y} = d(e_1, e_2). \tag{2}$$

This approach aims to predict the similarity score ($\hat{y}$) close to the ground-truth similarity score ($y$) by minimizing the mean squared error (MSE) which is given by:

$$\mathcal{L}(\theta_f) = \sum_{i=1}^{N} (\hat{y}_i - y_i)^2, \tag{3}$$

where $\theta_f$ is the parameter of embedding model $F$.

## D Limitations and Future Work

Although extensive experiments shows the effectiveness of our method, adjustment of the importance of each sentence may lead to learning a bias from the classifier. In future work, we plan to conduct an in-depth human analysis for machine-written sentences that are judged to be realistic or not. On the other hand, our work focused on unrealistic sentence in sentence similarity comparison tasks. The effect of training unrealistic examples in other natural language tasks worth to be explored. We will remain this analysis as our future work.

## E Detailed Analysis on Table 1

In this section, we present our detailed observations in Table 1 and the results of‘ the different N-gram in BLEU and Jaccard similarity. we observe that the number of unique N-gram occurs frequently when $p_D(x_m)$ is high. In terms of lexical similarity (BLEU and Jaccard) with a paired sentences, the scores of synthetic sentences $x_m$ with high $p_D(x_m)$ are higher about 20 points than those with low $p_D(x_m)$ and are similar to $x_h$. The distribution of word usage in generated sentences are also close to human-written sentences when predicted realistic score is high in two out of three datasets. Based on these observations, we confirm that even though the sentences are generated by the same machine in the same environment, there is a large variance in the extent to which the sentences are shifted from human-written sentences. Therefore, it is critical to handle the generated sentences carefully so that the model is not biased to the sentences that are very different from human-written sentences (*i.e.*, unrealistic samples).

## F Additional Results

We further compare our model trained on synthetic data against the following sentence encoders that are fine-tuned on natural language inference (NLI) dataset: Universal Sentence Encoder(USE) (Cer et al., 2018), InferSent (Conneau et al., 2017), sentence-BERT (Reimers and Gurevych, 2019), and sentence-RoBERTa. We also compare our trained model against the models that are not trained on human-annotated dataset, namely: GloVe (Pennington et al., 2014), BERT-CLS, sentence-BERT, sentence-RoBERTa. We present the results in Table 6 along with the results in Table 2.

As we can see, our model outperforms all the other baselines that are not trained on human-annotated dataset, and sometimes even better than the models trained on human-annotated dataset (*i.e.*, NLI). Our method contributes to improve the performance of models trained on synthetic dataset. These results support our assumption that adjusting loss of each machine-written sentence according to the importance would help in enhancing the reliability of the model and making it less biased by unrealistic machine-written sentences. Especially, We find that the magnitude of improvement is usually higher when the model is evaluated on the dataset which is not a source of human-written sentence $x_h$. These results imply that our method can enhance robustness of the sentence encoder with synthetic dataset when the sentence distribution is shifted. In terms of the variants of our method, using the randomly sampled scalar value as importance score usually degrades performance. In addition, filtering unrealistic examples without adjustment show lower performance than RISE. Based on these observations, we confirm that information about how realistic each example is contributes to the sentence encoder trained on synthetically generated datasets.

| | STSb | | | QQP | | | MRPC | | |
|---|---|---|---|---|---|---|---|---|---|
| | $x_h$ | $p_D(x_m)\uparrow$ | $p_D(x_m)\downarrow$ | $x_h$ | $p_D(x_m)\uparrow$ | $p_D(x_m)\downarrow$ | $x_h$ | $p_D(x_m)\uparrow$ | $p_D(x_m)\downarrow$ |
| BLEU-1 | 51.02 | 40.87 | 7.53 | 45.94 | 46.88 | 13.46 | 61.86 | 59.17 | 15.19 |
| BLEU-2 | 37.55 | 27.01 | 2.07 | 32.25 | 36.14 | 7.71 | 51.13 | 49.36 | 3.93 |
| BLEU-3 | 28.51 | 19.88 | 1.20 | 24.19 | 30.49 | 5.68 | 43.57 | 42.42 | 1.92 |
| BLEU-4 | 22.10 | 15.22 | 0.90 | 18.80 | 26.28 | 4.57 | 37.57 | 36.92 | 1.30 |
| BLEU-N | 34.80 | 25.75 | 2.93 | 30.3 | 34.95 | 7.86 | 48.53 | 46.97 | 5.59 |
| Jaccard | 41.98 | 33.97 | 5.98 | 39.91 | 42.49 | 11.31 | 53.55 | 53.33 | 10.52 |
| Distinct-1 | 8.5 | 5.1 | 1.8 | 5.7 | 3.7 | 3.4 | 7.8 | 4.3 | 2.5 |
| Distinct-2 | 49.7 | 36.5 | 15.0 | 39.5 | 25.5 | 23.4 | 48.7 | 31.4 | 20.1 |
| Distinct-3 | 75.4 | 66.2 | 34.3 | 69.1 | 46.5 | 45.5 | 77.4 | 60.6 | 43.4 |
| Distinct-N | 44.53 | 35.93 | 17.03 | 38.10 | 25.23 | 24.10 | 44.63 | 32.10 | 22.00 |
| Zipf coeff. | 1.03 | 1.07 | 1.23 | 1.11 | 1.06 | 1.12 | 0.98 | 1.02 | 1.23 |

Table 5: Results for comparing the sentences in different group. Jaccard indicates Jaccard similarity score. The score of generated sentences that is far from human scores is highlighted in underline. For BLEU-N and Distinct-N, we report the average score with different $N$.

| | | STSb | | QQP | | MRPC | | PAWS |
|---|---|---|---|---|---|---|---|---|
| $C_{src}$ | **Model** | $r$ | $\rho$ | Acc. | F1 | Acc. | F1 | F1 |
| | GloVe | 47.30 | 50.70 | 68.51 | 63.30 | 71.53 | 80.91 | 44.16 |
| | BERT-CLS | 17.18 | 20.30 | 66.38 | 61.50 | 66.03 | 79.79 | 49.32 |
| | BERT | 47.91 | 47.29 | 68.70 | 64.26 | 70.38 | 80.50 | 46.05 |
| | BERT* | 74.15 | 76.98 | 73.10 | 67.08 | 73.39 | 81.68 | 53.91 |
| | RoBERTa | 52.36 | 54.35 | 67.91 | 63.67 | 72.28 | 81.20 | 44.03 |
| | RoBERTa* | 74.78 | 77.80 | 73.56 | 67.00 | 75.76 | 82.46 | 56.48 |
| | USE* | 78.72 | 77.08 | 73.19 | 69.27 | 67.47 | 80.35 | 45.34 |
| | InferSent* | 49.53 | 50.86 | 68.94 | 64.13 | 65.97 | 79.32 | 45.01 |
| ***STSb*** | DINO | 78.45 | 77.71 | 73.14 | 68.04 | 70.44 | 81.16 | 47.30 |
| | RISE | **79.11** (+0.66) | **78.57** (+1.46) | **74.47** (1.33) | **69.08** (+1.04) | **72.84** (+2.4) | **82.01** (+0.85) | **50.24** (+2.94) |
| | ∟ Filtering | 77.73 (-0.72) | 77.45 (+0.34) | 73.06 (-0.08) | 67.94 (-0.10) | 68.96 (-1.48) | 81.35 (+0.19) | 46.72 (-0.58) |
| | ∟ Random | 79.03 (+0.58) | 78.39 (+1.28) | 73.09 (-0.05) | 68.03 (-0.01) | 71.09 (+0.65) | 81.62 (+0.46) | 50.17 (+2.87) |
| ***QQP*** | DINO | 64.93 | 65.93 | 73.20 | 67.72 | 70.75 | 80.40 | 44.47 |
| | RISE | **78.36** (+13.43) | **77.13** (+11.2) | 73.35 (+0.15) | 67.76 (+0.04) | **72.38** (+1.63) | **81.35** (+0.95) | 46.28 (+1.81 ) |
| | ∟ Filtering | 65.24 (+0.31) | 66.36 (+0.43) | **73.48** (+0.28) | **67.95** (+0.23) | 69.77 (-0.98) | 80.26 (-0.14) | 43.36 (-1.11) |
| | ∟ Random | 73.49 (+8.56) | 72.88 (+6.95) | 73.14 (-0.06) | 67.75 (+0.03) | 69.76 (-0.99) | 80.83( +0.43) | **46.97** (+2.5) |
| ***MRPC*** | DINO | 75.51 | 73.87 | 71.85 | 65.70 | 71.57 | 81.55 | 47.35 |
| | RISE | **77.47** (+1.96) | **76.86** (+2.99) | **74.23** (+2.38) | **68.82** (+3.12) | 71.97 (+0.4) | **81.95** (+0.4) | **49.35** (+2.00) |
| | ∟ Filtering | 76.25 (+0.74) | 74.88 (+1.01) | 71.05 (-0.80) | 64.82 (-0.88) | 71.34 (-0.23) | 80.76 (-0.79) | 47.84 (+0.49) |
| | ∟ Random | 76.06 (+0.55) | 74.51 (+0.64) | 72.52 (+0.67) | 66.45 (+0.75) | **72.19** (+0.62) | 81.71 (+0.16) | 47.56 (+0.21) |

Table 6: Evaluation results of different sentence embedding models on four sentence similarity task dataset. The models trained with human-annotated dataset (e.g., NLI) are marked with *. BERT and RoBERTa indicate sentence-BERT and sentence-RoBERTa, respectively. We highlight the best result in each pair of $C_{src}$/evaluation datasets and the best result in overall result in each metric as **bold** and underline, respectively. The number in right bracket indicates the performance difference with DINO. For regression task, we use Pearson correlation ($r$) and Spearman's rank correlation coefficient ($\rho$) metrics are used for evaluation. Each score represents the average of five trials.