# GAN-LM: Generative Adversarial Network using Language Models for Downstream Applications

*Dae Yon Hwang, Yaroslav Nechaev, Cyprien De Lichy, Renxian Zhang*

September 15, 2023

# Meet your host


**Dae Yon Hwang**
Applied Scientist, Amazon Alexa AI
https://eoduself.github.io/daeyonhwang





amazon | science

# Table of Contents

# Backgrounds #1

- Large corpora and computational resources have led to development of large language models (LLMs) ubiquitous in a wide variety of tasks.

- The performance loss in no- or low-resource settings can be substantial compared to their high-resource counterparts.

- A large amount of data is important to ensure the generalization of a model but it is not always possible due to cost and time constraints or lack of target language data, experts.

- Data augmentation (DA) can be a solution which allows to artificially increase the size of a dataset which ensures the generalization of a model.

# Backgrounds #2

- As a novel data generation, we propose ***Generative Adversarial Network using Language Models (GAN-LM).***

- Introduce tunable thresholds and a decoding method to control the diversity and lexical similarity of synthetic data to mitigate the mode collapse problem in GAN.

- GAN-LM employs an adversarial training with the offered data in each task to learn the different characteristic which generates suitable synthetic data for each task.

- Also, we mixed GAN-LM with other DAs (e.g. Back-translation) to enhance further in low-resource languages and limited entity linking task.

# Methodologies - Baseline #1

- Four different non-contextual-level augmentations are considered.

(1) Lexical: Use WordNet [1] to replace each word in the original text with a synonym.

(2) Spelling: Generate alternate texts from common misspellings of the original words [2].

(3) Character: Randomly change characters in the original tokens with four different ways: Insertions, substitutions, swaps and deletions [3].

(4) Token-LM: Use LM to get token for input text and then, perform nearest neighbor search for each token to find alternate tokens. BART [4] and mBART [5] are considered.


[1] Miller, George A. et al. "Introduction to WordNet: An On-line Lexical Database." International Journal of Lexicography 3 (1990): 235-244.
[2] Coulombe, Claude. "Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs." ArXiv abs/1812.04718 (2018): n. pag.
[3] Pruthi, Danish et al. "Combating Adversarial Misspellings with Robust Word Recognition." Annual Meeting of the Association for Computational Linguistics (2019).
[4] Lewis, Mike et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." Annual Meeting of the Association for Computational Linguistics (2019).
[5] Tang, Y. et al. "Multilingual Translation with Extensible Multilingual Pretraining and Finetuning." ArXiv abs/2008.00401 (2020): n. pag.


 

# Methodologies - Baseline #2

- Three different contextual-level augmentations are explored.

(1) Text Generation: Use the original text as the initial context and extend it. GPT-2 [6], OPT [7] and mGPT [8] are considered.

(2) Paraphrase: Transform a sentence with similar semantic meaning but a different syntactic form where T5 [9] and Prism model [10] are employed.

(3) Back-translation: Retranslate content from target language back to its source language to generate a sentence variant. Multiple pre-trained neural translation models are applied [11].

[6] Radford, Alec et al. “Language Models are Unsupervised Multitask Learners.” (2019).
[7] Zhang, Susan et al. “OPT: Open Pre-trained Transformer Language Models.” ArXiv abs/2205.01068 (2022): n. pag.
[8] Tan, Zhixing et al. “MSP: Multi-Stage Prompting for Making Pre-trained Language Models Better Translators.” ArXiv abs/2110.06609 (2021): n. pag.
[9] Raffel, Colin et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” ArXiv abs/1910.10683 (2019): n. pag.
[10] Thompson, Brian and Matt Post. “Automatic Machine Translation Evaluation in Many Languages via Zero-Shot Paraphrasing.” ArXiv abs/2004.14564 (2020): n. pag.
[11] Helsinki-NLP. 2023. Github - helsinki-nlp/opus-mt: Open neural machine translation models and web services.

# Methodologies - GAN-LM #1

- To extend the usability of GAN in NLP domain, we propose GAN-LM which combines GAN with pre-trained LM regardless of non-contextualized and contextualized models.

- We considered a WGAN-GP [12] which uses the Wasserstein distance as loss to capitalize on the probability distributions from fake and real data.

- Compared to the vanilla GAN, it is robust to vanishing gradient and mode collapse.

[12] Gulrajani, Ishaan et al. "Improved Training of Wasserstein GANs." NIPS (2017).

# Methodologies - GAN-LM #2

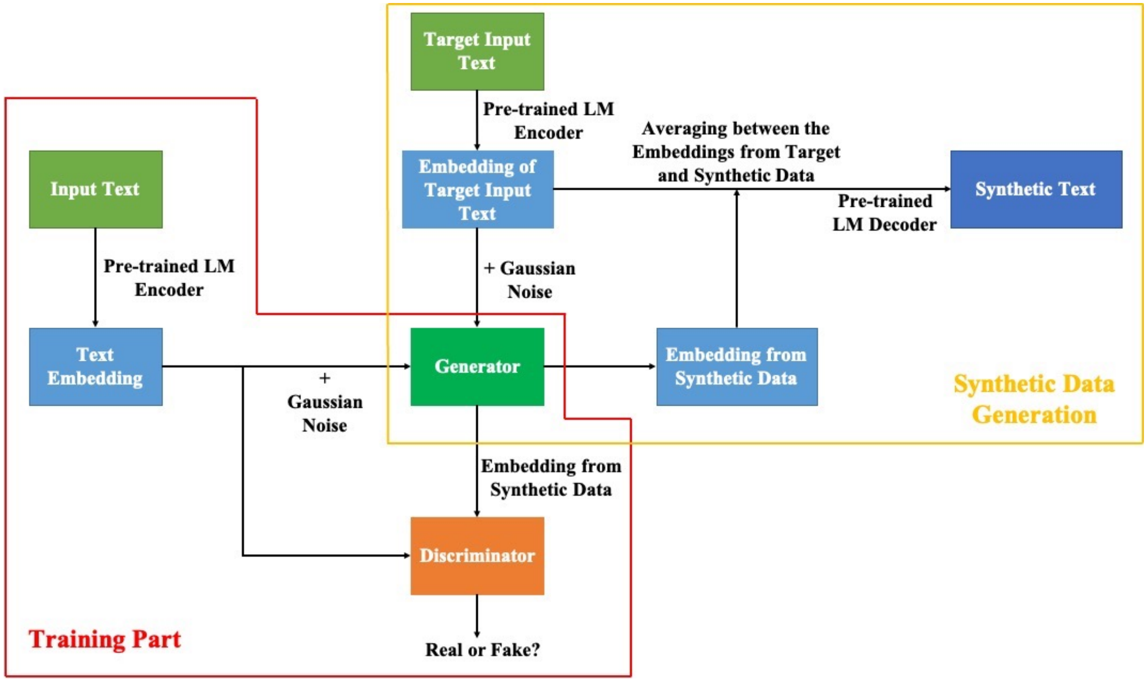


Figure 1. GAN-LM with pre-trained LM.

# Methodologies - GAN-LM #3

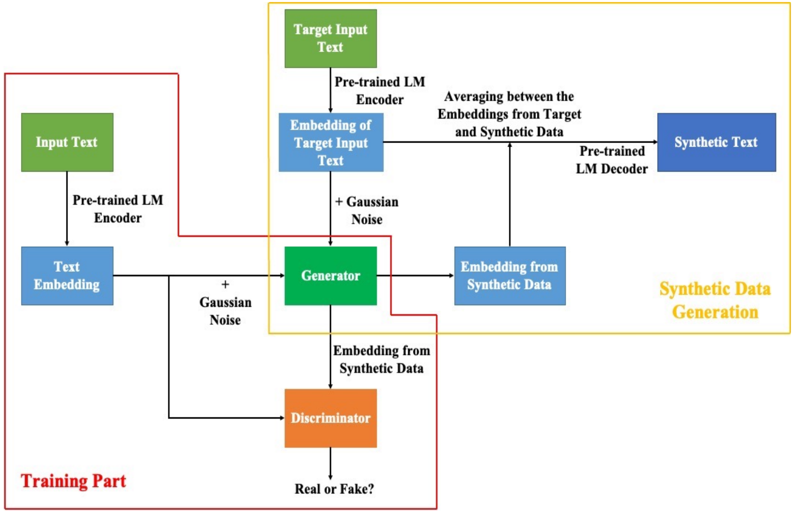


Figure 1. GAN-LM with pre-trained LM.

- In training part, we encode the input text into embeddings using pre-trained LM encoder.

- Then, we add Gaussian noise on top and input the resulting embeddings to the generator.

- Next, the generator produces synthetic embeddings which should resemble real ones.

- Lastly, we feed those to the discriminator which tries to distinguish between real and synthetic ones.

# Methodologies - GAN-LM #4

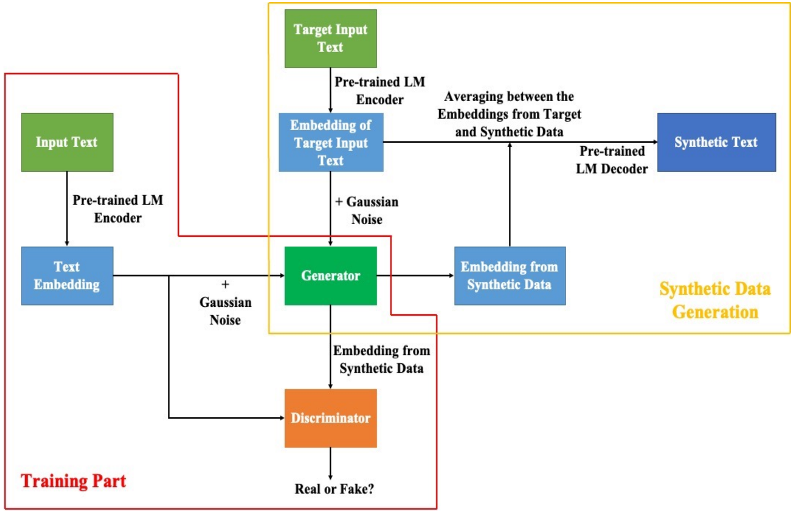


Figure 1. GAN-LM with pre-trained LM.

- In synthetic data generation, we feed the target text to the encoder and add Gaussian noise on it.

- The generator will produce the synthetic embedding for that target text.

- Then, we average the original and synthetic embeddings to maintain the structure of original text.

- To decode, we perform nearest neighbor search for each token using generated synthetic embeddings.

- Finally, we introduce similarity thresholds to find tokens that are diverse with similar semantics.

# Experimental Setting - Datasets and Employed Models

- (1) ZESHEL [13]: Zero-shot learning dataset for entity linking (EL) which is based on Wikia where there are non-overlapping domains in train/validation/test sets.

- (2) TREC [14]: Text retrieval dataset for question classification (QC) where questions were manually created with 50 fine class labels.

- (3) mSTS [15]: Multilingual version of semantic textual similarity (STS) task which has sentence pairs in 8 different languages.

[13] Logeswaran, Lajanugen et al. "Zero-Shot Entity Linking by Reading Entity Descriptions." ArXiv abs/1906.07348 (2019): n. pag.
[14] Li, Xin and Dan Roth. "Learning Question Classifiers." International Conference on Computational Linguistics (2002).
[15] Cer, Daniel Matthew et al. "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation." International Workshop on Semantic Evaluation (2017).

# Experimental Setting - More….

- For all downstream tasks, we construct a low-resource version (i.e. limited train set) to highlight augmentation impact.

- In EL task, ZESHEL contains rich textual context for both entity mentions and catalog entities. To isolate the impact of DA, we test model with and without those contexts.

- For EL task we used recall@k, for QC task F1 score, for STS task the spearman's rank correlation (SRC).

- In all experiments, we retrained target model 3 times with different seeds and reported average results with 95% confidence interval (CI).

# Results and Discussion - Entity Linking #1

Table 1. Recall values in ZESHEL without contexts.

| Scenarios | Type | R@1 | R@8 | R@32 | R@64 | CI | Change |
|---|---|---|---|---|---|---|---|
| Normal without context | **GAN-LM -GPT** | **28.91%** | **54.83%** | **64.77%** | **69.38%** | 1.71% | **7.94%** |
| | GAN-LM | 24.2% | 48.96% | 60.85% | 66.16% | **1.65%** | 3.51% |
| | GPT | 28.32% | 54.14% | 63.31% | 67.46% | 1.89% | 6.77% |
| | OPT | 27.54% | 53.28% | 62.81% | 67.15% | 1.89% | 6.16% |
| | Paraphrase | 22.1% | 46.89% | 59.1% | 64.73% | 2.03% | 1.67% |
| | Back-Translation | 20.7% | 44.77% | 57.13% | 62.99% | 2.06% | -0.14% |
| | Token-LM | 21.33% | 45.52% | 57.55% | 63.29% | 1.83% | 0.39% |
| | Char | 22.11% | 46.36% | 58.5% | 64.07% | 4.38% | 1.22% |
| | Spel | 21.52% | 45.76% | 58.22% | 63.88% | 2.25% | 0.81% |
| | Lexical | 20.67% | 44.8% | 57.23% | 62.91% | 2.01% | -0.13% |
| Low-resource without context | **GAN-LM -GPT** | **25.25%** | **50.94%** | **59.9%** | **63.8%** | 2.3% | **15.11%** |
| | GAN-LM | 18.67% | 42.43% | 55.21% | 61.03% | **1.97%** | 9.47% |
| | GPT | 22.52% | 47.52% | 58.23% | 62.62% | 2.37% | 12.86% |
| | OPT | 19.76% | 45.07% | 57.06% | 61.82% | 2.33% | 11.07% |
| | Paraphrase | 17.83% | 41.16% | 53.79% | 60% | 2.41% | 8.33% |
| | Back-Translation | 16.14% | 37.71% | 50.63% | 56.82% | 2.84% | 5.46% |
| | Token-LM | 15.86% | 36.9% | 49.98% | 56.2% | 2.9% | 4.87% |
| | Char | 16.52% | 37.91% | 51.34% | 57.53% | 2.67% | 5.96% |
| | Spel | 16.11% | 37.44% | 50.63% | 56.87% | 3.88% | 5.4% |
| | Lexical | 15.56% | 36.67% | 49.9% | 56.01% | 2.24% | 4.67% |
| | Baseline - Low | 12.4% | 31.24% | 44.65% | 51.16% | 3.09% | - |
| | Baseline - Normal | 20.57% | 44.89% | 57.56% | 63.13% | 1.92% | - |

- *Target:* Find the generalized augmentations for zero-shot learning task.

- There are large improvements, especially with contextual-level.

- GAN-LM mostly outperforms, except for GPT and OPT.

- In this case, EL model has been trained on only entity in train set to infer the entity with its contexts in test set.

- We further investigated the combination between GAN-LM and GPT, called GAN-LM-GPT.

- We observed improvements after combinations of both methods, especially in the low-resource case.

# Results and Discussion - Entity Linking #2

Table 2. Recall values in ZESHEL with contexts.

| Scenarios | Type | R@1 | R@8 | R@32 | R@64 | CI | Change |
|---|---|---|---|---|---|---|---|
| Normal with context | **GAN-LM** | **39.13%** | **66.45%** | **76.3%** | **79.98%** | **0.65%** | **1.23%** |
| | GPT | 37.36% | 65.31% | 74.78% | 78.65% | 1.54% | -0.21% |
| | OPT | 37.63% | 65.37% | 74.88% | 78.77% | 0.93% | -0.08% |
| | Paraphrase | 37.88% | 65.35% | 74.94% | 78.7% | 0.76% | -0.02% |
| | Back-Translation | 37.73% | 65.26% | 74.95% | 78.73% | 1.25% | -0.07% |
| | Token-LM | 37.53% | 64.58% | 74.49% | 78.41% | 1.27% | -0.49% |
| | Char | 37.53% | 64.68% | 74.6% | 78.56% | 1.37% | -0.4% |
| | Spel | 37.27% | 64.42% | 74.42% | 78.38% | 1.19% | -0.62% |
| | Lexical | 37.49% | 64.86% | 74.89% | 78.66% | 1.66% | -0.27% |
| Low-resource with context | **GAN-LM** | **23.93%** | **49.79%** | **61.5%** | **66.75%** | **1.29%** | **3.71%** |
| | GPT | 21.57% | 47.75% | 59.75% | 64.69% | 2.05% | 1.66% |
| | OPT | 22.84% | 47.99% | 60.47% | 65.38% | 1.68% | 2.39% |
| | Paraphrase | 20.13% | 45.59% | 58.36% | 63.62% | 1.75% | 0.14% |
| | Back-Translation | 17.6% | 42.25% | 54.86% | 60.84% | 1.98% | -2.9% |
| | Token-LM | 13.76% | 35.95% | 48.64% | 54.97% | 1.62% | -8.45% |
| | Char | 14.92% | 38.11% | 51.17% | 57.35% | 2.85% | -6.4% |
| | Spel | 19.46% | 44.46% | 56.85% | 62.54% | 4.71% | -0.96% |
| | Lexical | 17.59% | 41.68% | 54.03% | 60.18% | 2.62% | -3.41% |
| | Baseline - Low | 20.92% | 45.19% | 57.63% | 63.39% | 1.59% | - |
| | Baseline - Normal | 37.93% | 65% | 75.08% | 78.95% | 1.19% | - |

- For scenarios with context, most augmentations decrease the performance.

- This is because synthetic data is less related to the available contexts.

- However, GAN-LM always promises the improvements.

- We observed that GAN-LM and its complement, GAN-LM-GPT, are the best choices for entity linking task.

# Results and Discussion - Question Classification

Table 3. F1 values in TREC.

| Scenarios | Type | F1 | CI | Change |
|---|---|---|---|---|
| Half-train set | **GAN-LM** | **32.14%** | 2.23% | **16.01%** |
| | GPT | 29.16% | 2.66% | 13.03% |
| | OPT | 28.75% | 2.7% | 12.62% |
| | Paraphrase | 28.39% | 3% | 12.26% |
| | Back-Translation | 28.03% | 2.36% | 11.9% |
| | Token-LM | 27.16% | **1.67%** | 11.03% |
| | Char | 25.5% | 7.02% | 9.37% |
| | Spel | 29.05% | 2.16% | 12.92% |
| | Lexical | 26.93% | 5.02% | 10.8% |
| Low-resource | **GAN-LM** | **10.15%** | 1.95% | **9.27%** |
| | GPT | 8.48% | 3.61% | 7.6% |
| | OPT | 8.17% | 1.9% | 7.29% |
| | Paraphrase | 5.93% | 2.42% | 5.05% |
| | Back-Translation | 7.27% | 1.59% | 6.39% |
| | Token-LM | 5.26% | 3.72% | 4.38% |
| | Char | 4.19% | **1.42%** | 3.31% |
| | Spel | 7.68% | 4.03% | 6.8% |
| | Lexical | 6.09% | 3.3% | 5.21% |
| | Baseline - Low | 0.88% | 1.54% | - |
| | Baseline - Half | 16.13% | 1.16% | - |
| | Baseline - Normal | 24.97% | 2.27% | - |

- *Target:* Find label-invariant augmentations to improve the performance.

- Contextual-level augmentations mostly outperforms the non-contextual ones.

- GAN-LM is always the best performing approach which has 7.17% F1 improvement against Baseline - Normal.

# Results and Discussion - Multilingual STS #1

Table 4. SRC values in mSTS.

| Scenarios | Type | EN-AR | ES-EN | EN-DE | EN-TR | FR-EN | IT-EN | NL-EN | CI | Change |
|---|---|---|---|---|---|---|---|---|---|---|
| Normal | **GAN-LM -Back** | 46.18% | **55.92%** | **59.23%** | **43.72%** | 60.93% | **57.32%** | 53.9% | 2.64% | **2.38%** |
| | GAN-LM | 44.44% | 53.6% | 59.2% | 42.62% | **61.48%** | 55.31% | **53.96%** | **2.62%** | 1.43% |
| | mGPT | 45.24% | 50.86% | 59.2% | 42.52% | 60.51% | 53.07% | 53.86% | 2.71% | 0.67% |
| | Paraphrase | 45.21% | 48.69% | 58.06% | 40.9% | 60.67% | 54.12% | 53.32% | 2.92% | 0.06% |
| | Back-Translation | **46.36%** | 50.62% | 57.26% | 41.82% | 58.64% | 53.48% | 52.98% | 2.72% | 0.08% |
| Low-resource | **GAN-LM** | **31.75%** | **37.05%** | **44.71%** | **24.21%** | **43.12%** | **39.96%** | **43.96%** | **3.06%** | **5.43%** |
| | mGPT | 30.29% | 34.33% | 38.11% | 19.64% | 34.9% | 33.37% | 39.19% | 4.83% | 0.44% |
| | Paraphrase | 28.67% | 35.93% | 37.76% | 22.04% | 35.4% | 32.63% | 35.24% | 3.59% | 0.13% |
| | Back-Translation | 31.01% | 34.44% | 36.67% | 21.94% | 36.28% | 31.7% | 37.15% | 4.49% | 0.35% |
| | Baseline - Low | 29.95% | 33.13% | 36.04% | 18.23% | 37.26% | 34.68% | 37.46% | 3.85% | - |
| | Baseline - Normal | 45.08% | 50.52% | 56.9% | 40.94% | 60.89% | 53.16% | 53.08% | 2.47% | - |

- *Target:* Find diverse and semantically consistent augmented samples in multilingual.
- In low-resource, all augmentations improve the overall performance, especially with GAN-LM.
- Improvement in normal is lower but still, GAN-LM mostly gives the best results, except for EN-AR.

# Results and Discussion - Multilingual STS #2

Table 4. SRC values in mSTS.

| Scenarios | Type | EN-AR | ES-EN | EN-DE | EN-TR | FR-EN | IT-EN | NL-EN | CI | Change |
|---|---|---|---|---|---|---|---|---|---|---|
| Normal | **GAN-LM -Back** | 46.18% | **55.92%** | **59.23%** | **43.72%** | 60.93% | **57.32%** | 53.9% | 2.64% | **2.38%** |
| | GAN-LM | 44.44% | 53.6% | 59.2% | 42.62% | **61.48%** | 55.31% | **53.96%** | **2.62%** | 1.43% |
| | mGPT | 45.24% | 50.86% | 59.2% | 42.52% | 60.51% | 53.07% | 53.86% | 2.71% | 0.67% |
| | Paraphrase | 45.21% | 48.69% | 58.06% | 40.9% | 60.67% | 54.12% | 53.32% | 2.92% | 0.06% |
| | Back-Translation | **46.36%** | 50.62% | 57.26% | 41.82% | 58.64% | 53.48% | 52.98% | 2.72% | 0.08% |
| Low-resource | **GAN-LM** | **31.75%** | **37.05%** | **44.71%** | **24.21%** | **43.12%** | **39.96%** | **43.96%** | **3.06%** | **5.43%** |
| | mGPT | 30.29% | 34.33% | 38.11% | 19.64% | 34.9% | 33.37% | 39.19% | 4.83% | 0.44% |
| | Paraphrase | 28.67% | 35.93% | 37.76% | 22.04% | 35.4% | 32.63% | 35.24% | 3.59% | 0.13% |
| | Back-Translation | 31.01% | 34.44% | 36.67% | 21.94% | 36.28% | 31.7% | 37.15% | 4.49% | 0.35% |
| | Baseline - Low | 29.95% | 33.13% | 36.04% | 18.23% | 37.26% | 34.68% | 37.46% | 3.85% | - |
| | Baseline - Normal | 45.08% | 50.52% | 56.9% | 40.94% | 60.89% | 53.16% | 53.08% | 2.47% | - |

- GAN-LM is mostly trained on Indo-European languages (i.e. EN, DE, NL, FR, ES, IT) which enhances the generation ability for these languages.

- Back-translation works the best in EN-AR because it directly uses the well-defined translation models and this decreases the unsuitable assigned languages (e.g. code-switching).

- We combined GAN-LM with back-translation, GAN-LM-Back, to enhance further.

# Results and Discussion – Example of Augmented Data

Table 5. Examples of generated augmentations. Bold texts in each cell mean the changed parts.

| Type | Example |
|---|---|
| Original | Why do heavier objects travel downhill faster ? |
| Lexical | Why do heavier object travel downhill **quicker**? |
| Spelling | **Whay** do heavier objects travel downhill faster? |
| Character | Why do heavier **osbjects tralvel downhzill** faster? |
| Token-LM | **WHY does** heavier objects travel downhill faster ? |
| Back-Translation | Why **are the** heavier objects **moving down faster?** |
| Paraphrase | Why do heavier objects **go faster downhill?** |
| OPT | Why do heavier objects travel downhill faster ? **Because they're heavier** |
| GPT | Why do heavier objects travel downhill faster ? **Or slow down to 2 km h** |
| GAN-LM | **HOW** do heavier objects travel **down** faster ? |
| GAN-LM-GPT | **HOW** do heavier objects travel **down** faster ? **Or slow down to 2 km h** |





amazon science

# Conclusion

- In this work, we investigated the effect of different DAs to improve the performance on various tasks.

- We studied both techniques found in the literature as well as the proposed GAN-LM.

- We subsampled training sets to study model performance under low-resource conditions and used half or full training set to understand under different conditions.

- In most experiments, GAN-LM clearly gives the better results than non-contextual and contextual-level augmentations.

- In addition to apply GAN-LM solely, we combined it with GPT and back-translation to supplement the performance.

# Limitation

- There are three predictable limitations in the developed GAN-LM.

- (1) The convergence of training process in GAN-LM should be investigated carefully. We may need a few iterations of training to confirm the suitable epochs for each task.

- (2) There can be a machine bias since each downstream model is trained on machine generated synthetic data. Thus, searching the suitable pre-trained model is important.

- (3) GAN-LM is a general-purpose approach and its effectiveness on specific tasks or domains may vary even if we did a thorough evaluation on four downstream tasks.

# Thank You.

amazon | science

# Related Works #1 - Appendix

- There are relatively few works using GANs for text generation even if it is one of the most notable approaches in other domains.

- GAN model with Gumbel-Softmax was developed to have a differentiable sampling distribution for approximating a categorical one *.

- GANs with recurrent and convolutional architectures were developed for text augmentation at word and character-levels **.

- Sequence GAN with reinforcement learning was suggested to address the problem of assessing a partially generated sequence ***.

* Kusner, Matt J. and José Miguel Hernández-Lobato. “GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution.” ArXiv abs/1611.04051 (2016): n. pag.
** Subramanian, Sandeep et al. “Adversarial Generation of Natural Language.” ArXiv abs/1705.10929 (2017): n. pag.
*** Yu, Lantao et al. “SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient.” ArXiv abs/1609.05473 (2016): n. pag.

# Related Works #2 - Appendix

- Sequential GAN was explored as a data generation for the bootstrapping of a new language and the handling of low-resource features *.

- As far as we know, the work in ** was the first work to consider GAN with pre-trained LM (BERT) but it was mainly for reducing the time consumption of annotating the data.

- In ***, out-of-domain data generation with a sequential GAN was suggested to build the robust dialog system.

- GAN-LM combines LLM and GAN with tunable thresholds to suitably control the diversity and similarity of generated data. This extends the applicability to various tasks.

* Golovneva, O. Yu. and Charith S. Peris. “Generative Adversarial Networks for Annotated Data Augmentation in Data Sparse NLU.” ArXiv abs/2012.05302 (2020): n. pag.
** Croce, Danilo et al. “GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples.” Annual Meeting of the Association for Computational Linguistics (2020).
*** Marek, Petro et al. “OodGAN: Generative Adversarial Network for Out-of-Domain Data Generation.” ArXiv abs/2104.02484 (2021): n. pag.

# Backgrounds - Appendix

- GAN-LM employs an adversarial training with the offered data in each task to learn the different characteristic which generates suitable synthetic data for each task.

- Even if we used pre-trained LM in GAN-LM, we do not use its generation ability (e.g. paraphrase, summarization) for downstream tasks.

- Also, we mixed GAN-LM with other DAs (e.g. Back-translation) to enhance further in low-resource languages and limited entity linking task.

# Methodologies - What is GAN? - Appendix

- Generative Adversarial Network (GAN) is based on the adversarial learning which aims to trick the model by providing deceptive input.

- It consists of two neural networks, generator and discriminator, where each of them tries to outplay the other.

- The goal of generator is to manufacture outputs that could be hard to distinguish from real data. The discriminator aims to differentiate between real and synthetic data.

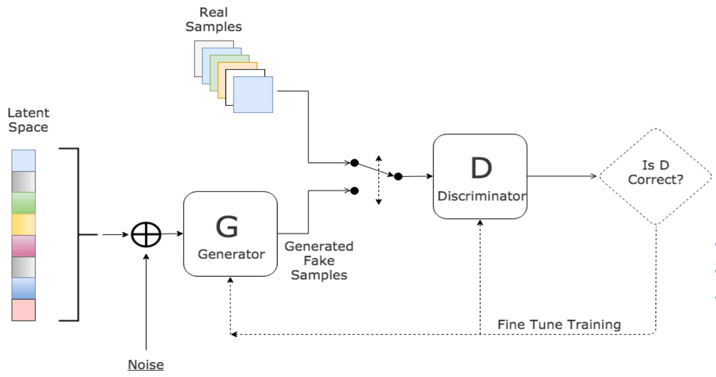


Figure 1. The introduction of GAN.

https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html

# Experimental Setting - Datasets and Employed Models - Appendix

- (1) ZESHEL *: Zero-shot learning dataset for entity linking (EL) which is based on Wikia where there are non-overlapping domains in train/validation/test sets. For this, we employed BLINK ** bi-encoder model from scratch.

- (2) TREC ***: Text retrieval dataset for question classification (QC) where questions were manually created with 50 fine class labels. For this application, we used fine-tuned BERT-Tiny **** with training data in TREC.

- (3) mSTS *****: Multilingual version of semantic textual similarity (STS) task which has sentence pairs in 8 different languages. For this task, we employed the mean pooling of the pre-trained multilingual BERT (mBERT) ****** with fine-tuning from train set.

* Logeswaran, Lajanugen et al. “Zero-Shot Entity Linking by Reading Entity Descriptions.” ArXiv abs/1906.07348 (2019): n. pag.
** Wu, Ledell Yu et al. “Zero-shot Entity Linking with Dense Entity Retrieval.” ArXiv abs/1911.03814 (2019): n. pag.
*** Li, Xin and Dan Roth. “Learning Question Classifiers.” International Conference on Computational Linguistics (2002).
**** Turc, Iulia et al. “Well-Read Students Learn Better: On the Importance of Pre-training Compact Models.” arXiv: Computation and Language (2019): n. pag.
***** Cer, Daniel Matthew et al. “SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation.” International Workshop on Semantic Evaluation (2017).
****** Devlin, Jacob et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” ArXiv abs/1810.04805 (2019): n. pag.

# Experimental Setting - Datasets and Employed Models - Appendix

- (3) STS-B *: Integrated version of semantic textual similarity (STS) task which includes news headlines, image captions and user forum posts.

For this task, we used SentenceTransformers ** from scratch using the mean pooling layer with the pre-trained XLM-RoBERTa ***.

- (4) mSTS *: Multilingual version of STS task which has sentence pairs in 8 different languages.

For this application, we employed the mean pooling of the pre-trained multilingual BERT (mBERT) **** with fine-tuning from train set.

* Cer, Daniel Matthew et al. "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation." International Workshop on Semantic Evaluation (2017).
** Reimers, Nils and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." Conference on Empirical Methods in Natural Language Processing (2019).
*** Conneau, Alexis et al. "Unsupervised Cross-lingual Representation Learning at Scale." Annual Meeting of the Association for Computational Linguistics (2019).
**** Devlin, Jacob et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." ArXiv abs/1810.04805 (2019): n. pag.

# Results and Discussion - Semantic Textual Similarity - Appendix

Table 4. SRC values in STS-B.

| Scenarios | Type | SRC | CI | Change |
|---|---|---|---|---|
| Half-train set | **GAN-LM** | **78.02%** | 0.96% | **4.44%** |
| | GPT | 76.94% | 0.83% | 3.36% |
| | OPT | 76.97% | 1.65% | 3.39% |
| | Paraphrase | 77.07% | 2.01% | 3.49% |
| | Back-Translation | 77.1% | 2.4% | 3.52% |
| | Token-LM | 76.11% | **0.57%** | 2.53% |
| | Char | 75.43% | 0.86% | 1.85% |
| | Spel | 76.61% | 2.13% | 3.03% |
| | Lexical | 76.74% | 1.39% | 3.16% |
| Low-resource | **GAN-LM** | **61.66%** | **1.46%** | **23.44%** |
| | GPT | 58.11% | 6.38% | 19.89% |
| | OPT | 59.17% | 3.95% | 20.95% |
| | Paraphrase | 57.9% | 3.1% | 19.68% |
| | Back-Translation | 58.02% | 6.72% | 19.8% |
| | Token-LM | 56.66% | 2.59% | 18.44% |
| | Char | 53.32% | 1.6% | 15.1% |
| | Spel | 54.52% | 5.07% | 16.3% |
| | Lexical | 57.77% | 5.17% | 19.55% |
| | Baseline - Low | 38.22% | 10.61% | - |
| | Baseline - Half | 73.58% | 4.08% | - |
| | Baseline - Normal | 78.49% | 0.28% | - |

- *Target:* Get various and semantically closed augmented data to improve the result.

- In low-resource, we could achieve great improvements, especially with contextual-level and GAN-LM.

- In half-train set, the improvement is smaller than the one in low-resource setting.

- Again, contextual-level outperforms non-contextual-level.

- GAN-LM yields the best performance which gives a closed performance as Baseline - Normal.





amazon | science