# Seeking Higher Truths and Higher Accuracies with Multilingual GAN-BERT

Stanford CS224N Custom Project

**James Thieu**
Department of Computer Science
Stanford University
jthieu23@stanford.edu

**Gil Kornberg**
Department of Computer Science
Stanford University
gilk@stanford.edu

**Charan Ramesh**
Stanford Center for Professional Development
Stanford University
crameshb@stanford.edu

## Abstract

Buddhist scriptures are often intentionally written to mirror the style of prior scriptures and quote prior texts verbatim. Moreover, the Buddhist canon is not uniform, split across many languages and schools. We therefore set out to design and build a model that accepts text from various languages and predicts the overall branch of Buddhism the text originates from, as well as the specific school of origin, formulated as two separate multi-class problems, respectively. In an effort to incorporate and improve upon state-of-the-art approaches in low-resource NLP tasks, we re-implemented and refined the GAN-BERT architecture to investigate methods of enhance finetuning for BERT. We also investigate the performance of standalone BERT, mBERT and LSTM models. We report that the LSTM model without pretrained embeddings obtains the highest accuracy on the 17-class classification task.

## 1 Key Information

- Sharing Project: Gil Kornberg is sharing the GAN architecture with CS236G project.
- External mentor: Dor Arad
- CS224N mentor: Angelica Sun

## 2 Introduction

Pre-training Language models using a large corpora is a proven methodology which improves the efficiency of the model on targeted tasks such as Classification, sentence completion, and question answering. Transformer based Architectures, eg:BERT have used this technique to achieve benchmark results. The model is primarily trained on a large corpus and further fine tuned on a smaller data-set focused on the desired task. In [1] the model is pre-trained on a large English corpus. The fine-tuning section is done for tasks solely in English Language. In this paper we utilize the BERT architecture to classify multilingual texts. This project is aimed at classifying multilingual Buddhist and other religious texts from various sources. The texts are from three main traditions of Buddhism representing the three main classes of this paper. The traditions are further divided into multiple schools from which the texts originated. Our prelimnary approach was to classify the texts into the three main classes (traditions). We further extended our approach by incorporating the subclasses (Schools of origin) and converting this into a multi-class (17-Classes) classification problem.

### 2.1 GANs in NLP

Generative Adversarial Networks (henceforth, GAN) have taken the computer vision community by storm in recent years. However, this success has not extended to applications in NLP due to the discreteness problem. Meanwhile, the use of pretrained BERT models as a basis for downstream tasks in NLP has obtained state of the art results in many NLP tasks including question answering and language inference, among others. There has been at least one attempt by Croce et. al. in GAN-BERT [2] to combine the robust language representation provided by BERT encodings with the propensity for GANs to generate realistic synthetic data. In GAN-BERT the authors had the intention of improving classification accuracy in low resource settings by leveraging fake BERT embeddings from the generator, with the final produce being the discriminator which doubles as a multi-class classifier. The purpose of this research is, in part, to extend the work done in GAN-BERT in two ways: to make the generator and discriminator more robust using transformer based architectures, and to evaluate the architecture's capacity to accommodate multilingual data and multilingual BERT. Results indicate that the transformer based GAN architecture is indeed more robust as it exhibits better results and more stable training when compared to the vanilla MLP-based model.

### 2.2 Challenges

The primary challenge in this task is the availability of data. The unorganized raw texts procured from multiple languages require considerable preprocessing. The structured data may not always be balanced across all classes in a classification problem. The over-representation of certain classes could cause adverse effects on the classifier. In this paper, we approach this problem using GAN. A GAN network comprises of a generator and a discriminator. The generator is provided with a specific data distribution based on which a fake distribution similar to that of the input is returned as the output. The discriminator then classifies the fake distributions from the true inputs. In this paper, we leverage the generated fake samples to normalize and fine tune the BERT.

Further, We explore the potential of BERT to exploit similarities and generalize across different languages. In [3], a multilingual neural model with a shared attention across all languages is discussed. The results from this method showed a dramatic improvement of performance on tasks executed on low-resource languages. Our dataset also includes texts from low-resource languages. Our BERT model is pre-trained using the multilingual data to observe the efficacy of the methodology discussed in [3].

## 3 Related Work

### 3.1 Multilingual Pre-training

The amount of data available in languages like English, French and German is significantly larger than resource scarce languages such as Pali or Sindhi. Hence, intuitively, the data limitation is expected to cause poor performance on low resource languages. Several researches have provided substantial evidences disproving this intuition. In [4], rather than normalizing all the available multilingual data, the entire dataset was trained on a single neural network with all constraints removed. The primary task of this paper was neural machine translation (NMT). The results show that, following the training of the complete corpora containing data from over 103 languages, the NMT of low-resource languages improved significantly. The generalizing capability of the model improved due to the linguistic and contextual similarities between the languages. Our intuition to pre-train BERT with multilingual data is derived from this work.

The usage of BERT architecture on multilingual data is described in [5]. The paper analyzes the extent to which the BERT model generalizes across languages. The result from [5] suggests that the model is able to exploit linguistic similarities between various languages and use the cross-lingual shared feature spaces to perform better on tasks such as Name Entity Recognition, translation and classification. Our core architecture is comprised of multilingual-BERT.

### 3.2 GANs for NLP

Several different approaches have been taken to try to generate synthetic natural language data using GANs. All suffer to varying degrees from problems of poor fidelity, lack of diversity, and mode

collapse. In SeqGAN [6], the researchers modelled the generator as a policy using a RL framework. This method is undesirable because it is often the case that models learn to exploit the reward functions and in doing so exhibit undesired behavior. LeakGAN [7] employs a similar approach. In another example, researchers attempted to use mixture adversarial generation in which multiple generators were trained, each with the aim of producing text of a different sentiment. In CatGAN [8], the authors enlisted the help of a category-aware model that directly measures the difference between real and generated samples for each category and uses this to compute the loss. Every approach that attempts to deal with words or sentences directly must contend with the challenge posed by the discreteness of text data. One partial solution to this problem is to use the Gumbel-Softmax activation function. The solution on which we are modelling our GAN approach is GANBERT [2], which involves training a generator to produce synthetic BERT embeddings. In this paper, the authors' aim is to improve BERT finetuning on classification tasks with limited training data by enlisting the help of the generator. However, in this approach the final product is the BERT model. We propose using a similar architecture with the final product being both the generator and the BERT model.

# 4 Approach

## 4.1 Baselines

### 4.1.1 BERT

Our first baseline was using a vanilla BERT to classify Mahayana English data according to the school that it belongs to. However, this was done because we had not finished collecting data, so we will consider our true first baseline to be using multilingual BERT on the fully compiled English-only data (this is discussed below).

This baseline was largely derived from [9], though we also added another output layer to predict labels as a multi-label problem. The model converts BERT sentence embeddings and attention masks to 768-dimension encodings per token, which we then pass through a Linear layer with 512 output dims. For the 3-class classification, this is fed through a 3-output Linear layer and softmax; for the multi-label problem, this is fed through a 16-output Linear layer and sigmoid activation.

The loss that results from this is an average loss between the negative log-likelihood (NLL) loss of the 3-way multi-class task and the Binary Cross-Entropy (BCE) loss of the 16-way multi-label task.

### 4.1.2 LSTM

Our first baseline is a basic LSTM model that largely uses the same architecture as in the BERT baseline; the major differences are that we replace the BERT model with an untrained LSTM and perform our own embedding. Rather than feed in 768-dimension BERT embeddings, we use an Embedding layer to generate 30-dimension BERT embeddings that accepts a maximum vocabulary size of 120,000, which is approximately the dimensionality of the multilingual BERT vocabulary [10]; this is because we keep the multilingual BERT tokenizer and merely use the sentence IDs, discarding the attention mask.

## 4.2 Switching From Multi-Label to Multi-Class

One of the issues with the multi-label problem is there are no convenient methods to have confusion matrices for one label against others; as such, this makes close analysis of confusion between different classes difficult without manual analysis. Our data is such that most of the datapoints have only one label, with tight clusters of correlated labels, as such, it was possible to reformulate the multi-label classification problem as a multi-class classification problem. For example, if a datapoint had the string of labels "Pure Land, Jodo-Shu, Jodo-Shinshu", the entire string would become a different class instead of 3 labels.

While this obviously raises the concern that results between the multi-label and multi-class classification tasks are not comparable, we believe that the greater interpretibility of results is more valuable.

Architecturally, this meant swapping out the 16-way output with sigmoid and associated BCE loss for a 17-way output with softmax.

### 4.3 GAN-BERT

We went about first re-implementing GAN-BERT in its original form. The blocks are comprised of a vanilla MLP generator with the following structure: each block is composed of a linear layer, a LeakyReLU, and a Dropout. The Generator is composed of three such blocks with a final linear layer which outputs a sequence length x embedding size tensor, a fake BERT embedding. The initial input is a noise tensor as per the GAN architecture. The Discriminator is composed of three blocks consisting of a linear layer with spectral norm initialization, followed by a LeakyReLU, and a final linear layer which outputs a single value. The training loop involves generating a fake embedding from the generator and a real embedding by passing a data point to the BERT model, and then taking the Least Squares Loss as described in LSGAN [11]:

$$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[(D(\mathbf{x})-b)^2] + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{data}(\mathbf{z})}\left[\left(D\left(G\left(\mathbf{z}\right)\right)-a\right)^2\right]$$

$$\min_G V_{LSGAN}\left(G\right) = \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}\left[\left(D\left(G\left(\mathbf{z}\right)\right)-c\right)^2\right]$$

where $a$ and $b$ are the labels for fake data and real data and $c$ denotes the value that $G$ wants $D$ to believe for fake data. Per convention, $a$ is a tensor of all zeros, $b$ is a tensor of all ones, and $c$ is a tensor of all ones. We experimented with BCE loss and Hinge loss as well, and least squares performed best as measured by training loss.

The augmented approach involved implementing novel generator and discriminator architectures as follows. The generator was replaced by a TransformerEncoder block comprised of six TransformerEncoderLayer blocks, each with the number of features equal to the embedding size. The discriminator was replaced by the same architecture with the addition of a final linear layer to output a single value. The only change made in the training loop is to generate a (batch size x sequence length x embedding size) noise tensor.

## 5 Experiments

### 5.1 Data

Our data is drawn from a variety of online databases of Buddhist texts, such as SuttaCentral and the 84000 project [12, 13]. We copied the text from various data formats into .txt files and then ran scripts & manually cleaned the data such that there is about 1 sentence of text per line (i.e. data point). This data is largely in English, though we also had text in Vietnamese, German, Indonesian, Classical Chinese, and modern Japanese.

Each data point is labeled with a Class field, which represents one of the 3 major branches of Buddhism (i.e. Mahayana, Vajrayana, Theravada) and a Label field, which represents the school(s) that hold a certain piece of text to be part of their school-specific canon. For example, the Lotus Sutra is a Mahayana sutra that is considered of principal importance by the Tendai and Nichiren schools of Buddhism. As such, a line from the Lotus Sutra would have both of those schools in the label field.

After scraping all of our data, we found that we had a large imbalance of the data, which was heavily weighted towards the Theravada class. For reference, of approx. 80,000 lines of text, around 50,000 were Theravada, with the remaining two classes evenly splitting the remaining 30,000 lines. To address this, we artificially generated a different version of the dataset that artificially constrained the number of Theravada samples.

We also generated versions of the dataset that had English-only data for all 3 branches (hereon referred to as AllEnglish). This version of the data has about 50,000 lines in total, with Mahayana and Theravada being slightly larger than the Vajrayana class. In addition, we had a version of the dataset that selectively incorporated about 3,000 lines of Mahayana multilingual data (hereon referred to as M_Augment) and another version which incorporated both the previous Mahayana multilingual data and 4,000 lines of Theravada multilingual data. Unfortunately, no multilingual Vajrayana data was available, so the inclusion of multilingual data would increase the class imbalance (hereon referred to as MT_Augment). Despite this, we wanted to see if the performance gain from the inclusion of the multilingual data could offset the consequences of this imbalance.

While we made efforts to make the distribution of the 3 Classes even, the distributions of data among the labels are extremely varied. This is also a subject of exploration. For additional exploration, we also ran some experiments on the raw, un-normalized data (hereon referred to as RawData).

### 5.2 Evaluation method

Our three evaluation metrics are precision, recall, and F1 score, which are typical for multi-class classification problems. Each class generates its own values for the three aforementioned scores, which we then weight based on the proportion of overall samples per class to formulate overall weighted average scores for each metric.

However, given the nature of the data, we will focus especially on model the Mahayana school data, as that is the only subset where datapoints have multiple labels. We will be looking closely at the confusion matrices among the aforementioned Mahayana schools.

### 5.3 Experimental details

We ran the BERT multilingual models on the AllEnglish, M_Augment and MT_Augment datasets. For both the model and tokenizer, we used the 'bert-multilingual-cased' pre-trained models [4].

The main hyperparameters that we focused on were batch size and the weighting between the loss functions.

Generally, the fixed hyperparameters were as follows:

- Epochs: 20
- Learning Rate: 3e-4
- Train-Val-Test Split: 80/10/10
- Max Sequence Length: 100

### 5.4 Results

Surprisingly, we find that the LSTM model generally outperformed the pretrained multilingual BERT. During training, we also observed that LSTM models took far less time to train - often, the time difference would be an entire order of magnitude between tens of minutes and several hours.

For the 3-class problem, we found that our models could achieve 80-90% weighted F1 scores, which is to be expected from such advanced models.

For the tables below, we only present the model results for the 17-way multi-class classification problem, as it is a more difficult and interesting problem than the 3-way version.

Table 1: Weighted Performance of mBERT vs LSTM

| Model | Dataset | Weighted Precision | Weighted Recall | Weighted F1 |
|---|---|---|---|---|
| mBERT | AllEnglish | .6329 | 0.5572 | 0.56 |
| LSTM | AllEnglish | 0.6305 | 0.5884 | 0.5982 |
| mBERT | M_Augment | 0.6424 | 0.6078 | 0.611 |
| LSTM | M_Augment | 0.707 | 0.656 | 0.6705 |
| mBERT | MT_Augment | 0.7861 | 0.4384 | 0.5334 |
| LSTM | MT_Augment | 0.6784 | 0.632 | 0.6399 |
| mBERT | RawData | 0.7623 | 0.6941 | 0.709 |
| LSTM | RawData | **0.797** | **0.7738** | **0.783** |
| GAN-mBERT | RawData | 0.628 | 0.546 | 0.563 |
| TransGAN-BERT | RawData | 0.655 | 0.546 | 0.567 |

Table 2: F1 Scores for 17-Way Multi-Class Classification

| Label | LSTM | mBERT | LSTM | mBERT |
|---|---|---|---|---|
| Data | RawData | RawData | M_Augment | M_Augment |
| General Mahayana | **0.405** | 0.3131 | 0.3354 | 0.3721 |
| Zen | 0.604 | **0.6457** | 0.5921 | 0.5524 |
| Tendai | **0.808** | 0.6611 | 0.7058 | 0.6694 |
| Nichiren | **0.406** | 0.3568 | 0.3864 | 0.3992 |
| General Vajrayana | 0.592 | 0.4995 | **0.6008** | 0.5112 |
| Shingon | **0.475** | 0.4050 | 0.4256 | 0.33 |
| Action Tantras | 0.2476 | 0.1834 | **0.3431** | 0.3140 |
| Dedication | 0.0 | 0.0 | 0.0 | 0.0 |
| Incantations | 0.371 | 0.3768 | 0.3682 | **0.3824** |
| Yoga Tantras | 0.2486 | **0.4305** | 0.3188 | 0.3961 |
| Abhidhamma | .9501 | 0.8624 | **0.9789** | 0.9097 |
| Sutta | **0.9099** | 0.8291 | 0.7888 | 0.6485 |
| Vinaya | 0.85 | 0.7400 | **0.8629** | 0.7428 |
| Tendai, Nichiren | **0.685** | 0.6037 | 0.6691 | 0.6697 |
| Pure Land, Jodo-Shu, Jodo-Shinshu | 0.374 | 0.4301 | **0.6654** | 0.6653 |
| Pure Land, Jodo-Shu | 0.5247 | **0.6318** | 0.6013 | 0.5129 |
| Pure Land, Jodo-Shinshu | **0.736** | 0.6131 | 0.4347 | 0.5269 |

# 6 Analysis

## 6.1 Multilingual Data

Based on our results, we found that the incorporation of multilingual data does have the potential to improve the performance of model. For our tests with and without the addition of foreign-language data, we found that adding the extra lines to Zen and Pure Land classes led to a 20-40% increase in their overall class F1 score. This additional data compensates for these classes being under-resourced.

However, when we added additional foreign-language data to the already well-resourced Theravada class (and its subclasses), there were no appreciable gains in performance; this indicates that the usage of multilingual data should be done in such a way that additional data is treated as language-agnostic, at which point the class balances become more important.

## 6.2 LSTM vs mBERT

Generally, we found that the LSTM models performed significantly better than the mBERT models when run under the same experimental conditions. This is likely due to the differing initialization conditions for the two - whereas mBERT was trained for the 100 languages using the entirety of those respective languages' Wikipedia's [4], the LSTM embedding layer was initialized from scratch. As a result, it is likely that, given the domain-specific vocabulary and usage of terms in different contexts, the randomly initialized embedding was better able to achieve embeddings that capture the granular usage of the vocabulary. For many of the labels, the differences perhaps did not require very complex representations; with the mBERT, complex representations likely had many artifacts from other usages that was impeding the convergence.

When we take into consideration that the LSTM embedding was only of size 30 while the mBERT embedding is of size 768, this idea is more plausible in the sense that movement of all 768 elements would be done much more slowly than shifting 30-long vectors.

### 6.3 Label Confusion

In the context of the dataset, one of the principal areas of interest was looking at the overlap between the labels "Pure Land, Jodo-Shu, Jodo-Shinshu", "Pure Land, Jodo-Shu", and "Pure Land, Jodo-Shinshu", which represent the wider Mahayana Pure Land school, its Japanese version (Jodo-Shu), and a later derivative branch of Jodo-Shu called Jodo-Shinshu, respectively.

While the performance for these 3 classes was relatively poor compared to the other classes with higher support, they actually perform very well with respect to each other.

Table 3: Confusion Matrix - LSTM (RawData)

| True Down / Predictions Right | Pure Land | Jodo-Shu | Jodo-Shinshu |
|---|---|---|---|
| Pure Land | 49 | 10 | 2 |
| Jodo-Shu | 7 | 85 | 4 |
| Jodo-Shinshu | 0 | 2 | 39 |

Looking at other parts of the confusion matrices, we noted that many documents were being misclassified as either General Mahayana (Label 0) or as General Vajrayana (Label 1). While the misclassification is disappointing, it indicates that the models, without ability to discern the fine-grain differences in text, will "default" to the more general classifications of text.

Another interesting result is that the 17 classes, with respect to confusions, tend to fit "snugly" into their 3 branch splits. This is likely due to the addition of the loss term for the 3-class task, which is trained to know that these schools belong together, so punishes the model for having the fine-grain labels which belong to a school "drift" to other schools. For example, Yoga Tantras, which had a very poor F1 score of 0.24, had most of its confusion with other datapoints that shared the Vajrayana label.

### 6.4 GAN-BERT

The GAN augmented BERT has obtained some improvement in performance consistent with the results of the original GAN-BERT paper, but not as much as we had hoped. Interestingly, although there are some non-English words in the data, the performance of the GAN augmented mBERT does not surpass that of the GAN augmented BERT. The attempt to decode the generated embeddings was unsuccessful as it did not yield high quality, synthetic sentences:

Table 4: Sample Generated Sentences

| |
|---|
| indeed to rather and and and |
| or or according or or or |
| and and and and and and |
| night, and,, with with |
| and. nor or nor never |
| proper certainly certainly especially especially |
| person most especially with generally |

It is encouraging that the generator does not seem to have mode collapsed completely, though the results do certainly leave something to be desired.

It was apparent from the loss curves that both the generator and discriminator losses converged relatively quickly. This doesn't necessarily indicate a problem, but it may suggest that the model could have required a significantly longer training regimen. Because the training loop involved propagating losses to and updating weights of three separate models, each of which had a considerably large number of parameters, training the GAN took a significant amount of time and compute. A longer training regimen might allow the generator to traverse the modes of the data distribution more broadly and avoid mode collapse.

## 7 Conclusion

With very domain-specific vocabularies, such as those of Buddhist scriptures, relying on pre-trained models readily available online may lead to worse results than just fine-tuning an embedding space from scratch, as that allows one to capture the differences in the smaller subspace without artifacting from other contexts. As such, if we had the time to pretrain our own BERT on domain specific data rather than relying on the BERT-base model this would likely have improved downstream accuracy significantly.

Additionally, when solving problems in fields with resources scattered across different languages, the usage of multi-lingual models should first bring up the question of whether a language is supported; if so, then data from different languages should be considered without special attention to the source language, with more emphasis being placed on the traditional dataset considerations such as token length, data cleaning, and dataset imbalances. However, we recommend that further work be done to see if using "edge" languages, such as classical Chinese and medieval Japanese, affects embeddings on these languages which are trained on modern descendant languages.

The GAN-BERT augmentation was encouraging because we successfully proved the superiority of a Transformer based generator and discriminator over the vanilla MLP. This is not surprising, as Transformers are more well suited to tasks in NLP, but it is a modest, novel contribution. In terms of future work, there are several intriguing directions to pursue. In terms of the loss function, it would be interesting to see how a PathGAN [14] discriminator would improve performance. Intuitively, it does not make sense that the discriminator should output a single scalar indicating how real or fake an embedding is. It makes more sense to discriminate across the entire embedding and output a matrix of real vs. fake as in the PatchGAN discriminator. In addition, a U-Net architecture might also be an interesting direction to pursue, Given that the smaller LSTM model outperformed the bigger BERT embeddings. Since the U-Net [14] design aims to downsample an image and extract the most important information, and with the understanding that the much smaller LSTM embeddings outperform BERT, this may be an indication that it is worthwhile to probe how we might increase the performance of the BERT model by extracting only the most salient information. Finally, considering the surprising performance of the LSTM model, it would be exciting to see if a GAN-LSTM could improve it even further.

## References


[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[2] Danilo Croce, Giuseppe Castellucci, and Roberto Basili. GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2119, Online, July 2020. Association for Computational Linguistics.

[3] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California, June 2016. Association for Computational Linguistics.

[4] Exploring massively multilingual, massive neural machine translation.

[5] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July 2019. Association for Computational Linguistics.

[6] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. *SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient*, 2016.

[7] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. *Long Text Generation via Adversarial Training with Leaked Information*, 2017.

[8] Zhiyue Liu, Jiahai Wang, and Zhiwei Liang. *CatGAN: Category-aware Generative Adversarial Networks with Hierarchical Evolutionary Learning for Category Text Generation*, 2019.

[9] Baseline Code: transfer learning for nlp fine-tuning bert for text classification. `https://www.analyticsvidhya.com/blog/2020/07/transfer-learning-for-nlp-fine-tuning-bert-for-text-classification/`,.

[10] Load what you need: Smaller versions of multilingual BERT. pages 119–123. Association for Computational Linguistics.

[11] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. *Least Squares Generative Adversarial Networks*, Oct 2017.

[12] Suttacentral. `https://suttacentral.net/`. Accessed: 2021-03-13.

[13] 84000 project. `https://84000.co/`. Accessed: 2021-03-13.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. *Image-to-Image Translation with Conditional Adversarial Networks*, Jul 2017.