GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples

Danilo Croce
Dept. of Enterprise Engineering
University of Rome, Tor Vergata
Roma, Italy
croce@info.uniroma2.it

Giuseppe Castellucci
Amazon
Seattle, USA
giusecas@amazon.com

Roberto Basili
Dept. of Enterprise Engineering
University of Rome, Tor Vergata
Roma, Italy
basili@info.uniroma2.it

Abstract

Recent Transformer-based architectures, e.g., BERT, provide impressive results in many Natural Language Processing tasks. However, most of the adopted benchmarks are made of (sometimes hundreds of) thousands of examples. In many real scenarios, obtaining high-quality annotated data is expensive and time-consuming; in contrast, unlabeled examples characterizing the target task can be, in general, easily collected. One promising method to enable semi-supervised learning has been proposed in image processing, based on Semi-Supervised Generative Adversarial Networks. In this paper, we propose GAN-BERT that extends the fine-tuning of BERT-like architectures with unlabeled data in a generative adversarial setting. Experimental results show that the requirement for annotated examples can be drastically reduced (up to only 50-100 annotated examples), still obtaining good performances in several sentence classification tasks.

1 Introduction

In recent years, Deep Learning methods have become very popular in Natural Language Processing (NLP), e.g., they reach high performances by relying on very simple input representations (for example, in (Kim, 2014; Goldberg, 2016; Kim et al., 2016)). In particular, Transformer-based architectures, e.g., BERT (Devlin et al., 2019), provide representations of their inputs as a result of a pre-training stage. These are, in fact, trained over large scale corpora and then effectively fine-tuned over a targeted task achieving state-of-the-art results in different and heterogeneous NLP tasks. These achievements are obtained when thousands of annotated examples exist for the final tasks. As experimented in this work, the quality of BERT fine-tuned over less than 200 annotated instances shows significant drops, especially in classification tasks involving many categories. Unfortunately, obtaining annotated data is a time-consuming and costly process. A viable solution is adopting semi-supervised methods, such as in (Weston et al., 2008; Chapelle et al., 2010; Yang et al., 2016; Kipf and Welling, 2016) to improve the generalization capability when few annotated data is available, while the acquisition of unlabeled sources is possible.

One effective semi-supervised method is implemented within Semi-Supervised Generative Adversarial Networks (SS-GANs). Usually, in GANs (Goodfellow et al., 2014) a "generator" is trained to produce samples resembling some data distribution. This training process "adversarially" depends on a "discriminator", which is instead trained to distinguish samples of the generator from the real instances. SS-GANs (Salimans et al., 2016) are an extension to GANs where the discriminator also assigns a category to each example while discriminating whether it was automatically generated or not.

In SS-GANs, the labeled material is thus used to train the discriminator, while the unlabeled examples (as well as the ones automatically generated) improve its inner representations. In image processing, SS-GANs have been shown to be effective: exposed to few dozens of labeled examples (but thousands of unlabeled ones), they obtain performances competitive with fully supervised settings.

In this paper, we extend the BERT training with unlabeled data in a generative adversarial setting. In particular, we enrich the BERT fine-tuning process with an SS-GAN perspective, in the so-called GAN-BERT[1] model. That is, a generator produces "fake" examples resembling the data distribution, while BERT is used as a discriminator. In this way, we exploit both the capability of BERT to produce high-quality representations of input texts and to adopt unlabeled material to help the network in

[1]The code is available at https://github.com/crux82/ganbert.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2114–2119
July 5 - 10, 2020.

generalizing its representations for the final tasks. At the best of our knowledge, using SS-GANs in NLP has been investigated only by (Croce et al., 2019) with the so-called Kernel-based GAN. In that work, authors extend a Kernel-based Deep Architecture (KDA, (Croce et al., 2017)) with an SS-GAN perspective. Sentences are projected into low-dimensional embeddings, which approximate the implicit space generated by using a Semantic Tree Kernel function. However, it only marginally investigated how the GAN perspective could extend deep architecture for NLP tasks. In particular, a KGAN operates in a pre-computed embedding space by approximating a kernel function (Annesi et al., 2014). While the SS-GAN improves the quality of the Multi-layered Perceptron used in the KDA, it does not affect the input representation space, which is statically derived by the kernel space approximation. In the present work, all the parameters of the network are instead considered during the training process, in line with the SS-GAN approaches.

We empirically demonstrate that the SS-GAN schema applied over BERT, i.e., `GAN-BERT`, reduces the requirement for annotated examples: even with less than 200 annotated examples it is possible to obtain results comparable with a fully supervised setting. In any case, the adopted semi-supervised schema always improves the result obtained by BERT.

In the rest of this paper, section 2 provides an introduction to SS-GANs. In sections 3 and 4, `GAN-BERT` and the experimental evaluations are presented. In section 5 conclusions are derived.

2 Semi-supervised GANs

SS-GANs (Salimans et al., 2016) enable semi-supervised learning in a GAN framework. A discriminator is trained over a $(k+1)$-class objective: "true" examples are classified in one of the target $(1, ..., k)$ classes, while the generated samples are classified into the $k+1$ class.

More formally, let $\mathcal{D}$ and $\mathcal{G}$ denote the discriminator and generator, and p_d and $p_{\mathcal{G}}$ denote the real data distribution and the generated examples, respectively. In order to train a semi-supervised k-class classifier, the objective of $\mathcal{D}$ is extended as follows. Let us define $p_m(\hat{y} = y|x, y = k+1)$ the probability provided by the model m that a generic example x is associated with the fake class and $p_m(\hat{y} = y|x, y \in (1, ..., k))$ that x is considered real, thus belonging to one of the target classes. The loss function of $\mathcal{D}$ is defined as: $L_{\mathcal{D}} = L_{\mathcal{D}_{\text{sup.}}} + L_{\mathcal{D}_{\text{unsup.}}}$ where:

$$L_{\mathcal{D}_{\text{sup.}}} = -\mathbb{E}_{x,y\sim p_d} \log[p_{\text{m}}(\hat{y} = y|x, y \in (1, ..., k))]$$

$$L_{\mathcal{D}_{\text{unsup.}}} = -\mathbb{E}_{x\sim p_d} \log[1 - p_{\text{m}}(\hat{y} = y|x, y = k+1)] - \mathbb{E}_{x\sim \mathcal{G}} \log[p_{\text{m}}(\hat{y} = y|x, y = k+1)]$$

$L_{\mathcal{D}_{\text{sup.}}}$ measures the error in assigning the wrong class to a real example among the original k categories. $L_{\mathcal{D}_{\text{unsup.}}}$ measures the error in incorrectly recognizing a real (unlabeled) example as fake and not recognizing a fake example.

At the same time, $\mathcal{G}$ is expected to generate examples that are similar to the ones sampled from the real distribution p_d. As suggested in (Salimans et al., 2016), $\mathcal{G}$ should generate data approximating the statistics of real data as much as possible. In other words, the *average* example generated in a batch by $\mathcal{G}$ should be similar to the real *prototypical* one. Formally, let's $f(x)$ denote the activation on an intermediate layer of $\mathcal{D}$. The *feature matching* loss of $\mathcal{G}$ is then defined as:

$$L_{\mathcal{G}_{\text{feature matching}}} = \left\| \mathbb{E}_{x \sim p_d} f(x) - \mathbb{E}_{x \sim \mathcal{G}} f(x) \right\|_2^2$$

that is, the generator should produce examples whose intermediate representations provided in input to $\mathcal{D}$ are very similar to the real ones. The $\mathcal{G}$ loss also considers the error induced by fake examples correctly identified by $\mathcal{D}$, i.e.,

$$L_{\mathcal{G}_{unsup.}} = -\mathbb{E}_{x\sim \mathcal{G}} \log[1 - p_m(\hat{y} = y|x, y = k+1)]$$

The $\mathcal{G}$ loss is $L_{\mathcal{G}} = L_{\mathcal{G}_{\text{feature matching}}} + L_{\mathcal{G}_{unsup.}}$.

While SS-GANs are usually used with image inputs, we will show that they can be adopted in combination with BERT (Devlin et al., 2019) over inputs encoding linguistic information.

3 GAN-BERT: Semi-supervised BERT with Adversarial Learning

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) belongs to the family of the so-called transfer learning methods, where a model is first pre-trained on general tasks and then fine-tuned on the final target tasks. In Computer Vision, transfer learning has been shown beneficial in many different tasks, i.e., pre-training a neural network model on a known task, followed by a fine-tuning stage on a (different) target task (see, for example, (Girshick et al., 2013)). BERT

is a very deep model that is pre-trained over large corpora of raw texts and then is fine-tuned on target annotated data. The building block of BERT is the *Transformer* (Vaswani et al., 2017), an attention-based mechanism that learns contextual relations between words (or sub-words, i.e., word pieces, (Schuster and Nakajima, 2012)) in a text.

BERT provides contextualized embeddings of the words composing a sentence as well as a *sentence embedding* capturing sentence-level semantics: the pre-training of BERT is designed to capture such information by relying on very large corpora. After the pre-training, BERT allows encoding (i) the words of a sentence, (ii) the entire sentence, and (iii) sentence pairs in dedicated embeddings. These can be used in input to further layers to solve sentence classification, sequence labeling or relational learning tasks: this is achieved by adding task-specific layers and by fine-tuning the entire architecture on annotated data.

In this work, we extend BERT by using SS-GANs for the fine-tuning stage. We take an already pre-trained BERT model and adapt the fine-tuning by adding two components: i) task-specific layers, as in the usual BERT fine-tuning; ii) SS-GAN layers to enable semi-supervised learning. Without loss of generality, let us assume we are facing a sentence classification task over k categories. Given an input sentence $s = (t_1, ..., t_n)$ BERT produces in output $n + 2$ vector representations in R^d, i.e., $(h_{CLS}, h_{t_1}, ..., h_{t_n}, h_{SEP})$. As suggested in (Devlin et al., 2019), we adopt the h_{CLS} representation as a sentence embedding for the target tasks.

As shown in figure 1, we add on top of BERT the SS-GAN architecture by introducing i) a discriminator $\mathcal{D}$ for classifying examples, and ii) a generator $\mathcal{G}$ acting adversarially. In particular, $\mathcal{G}$ is a Multi Layer Perceptron (MLP) that takes in input a 100-dimensional noise vector drawn from $N(\mu, \sigma^2)$ and produces in output a vector $h_{fake} \in R^d$. The discriminator is another MLP that receives in input a vector $h_* \in R^d$; h_* can be either h_{fake} produced by the generator or h_{CLS} for unlabeled or labeled examples from the real distribution. The last layer of $\mathcal{D}$ is a softmax-activated layer, whose output is a $k + 1$ vector of logits, as discussed in section 2.

During the forward step, when real instances are sampled (i.e., $h_* = h_{CLS}$), $\mathcal{D}$ should classify them in one of the k categories; when $h_* = h_{fake}$, it should classify each example in the $k + 1$ category. As discussed in section 2, the training process tries

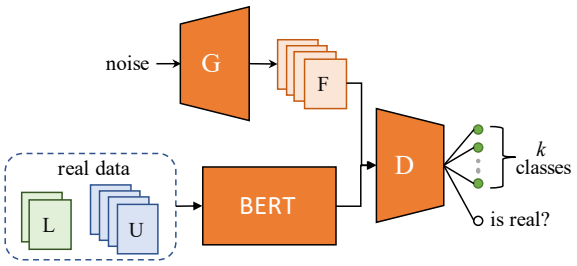


Figure 1: GAN-BERT architecture: $\mathcal{G}$ generates a set of fake examples F given a random distribution. These, along with unlabeled U and labeled L vector representations computed by BERT are used as input for the discriminator $\mathcal{D}$.

to optimize two competing losses, i.e., L_D and L_G.

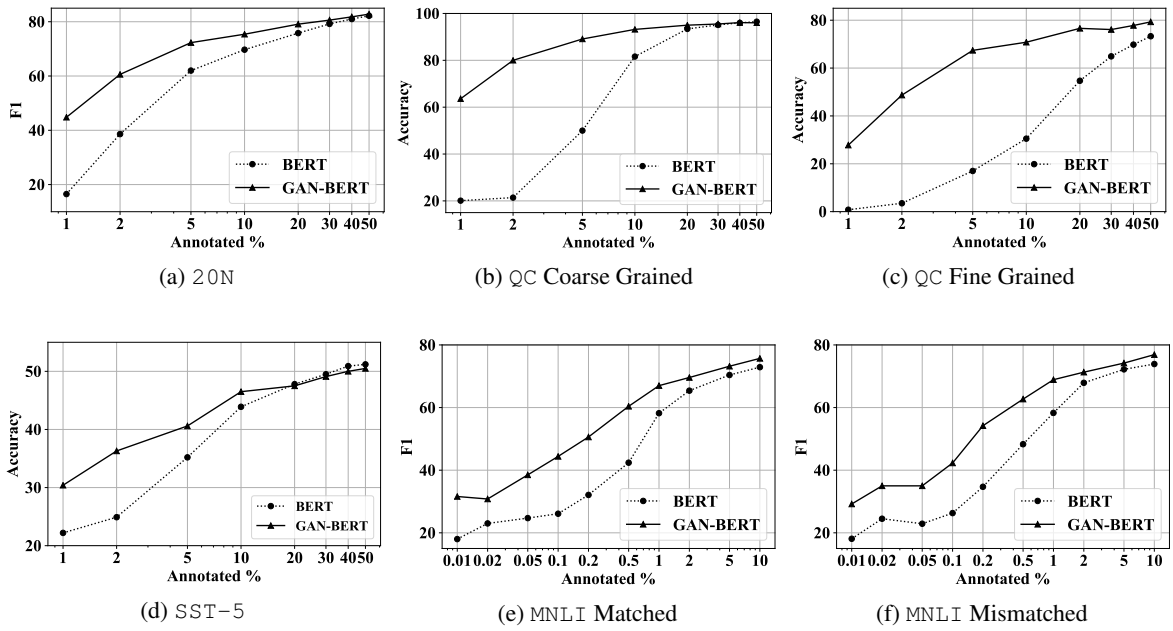
During back-propagation, the unlabeled examples contribute only to $L_{D_{unsup.}}$, i.e., they are considered in the loss computation only if they are erroneously classified into the $k+1$ category. In all other cases, their contribution to the loss is masked out. The labeled examples thus contribute to the supervised loss $L_{D_{sup.}}$. Finally, the examples generated by $\mathcal{G}$ contribute to both L_D and L_G, i.e., $\mathcal{D}$ is penalized when not finding examples generated by $\mathcal{G}$ and vice-versa. When updating $\mathcal{D}$, we also change the BERT weights in order to fine-tune its inner representations, so considering both the labeled and the unlabeled data[2].

After training, $\mathcal{G}$ is discarded while retaining the rest of the original BERT model for inference. This means that there is no additional cost at inference time with respect to the standard BERT model. In the following, we will refer to this architecture as GAN-BERT.

4 Experimental Results

In this section, we assess the impact of GAN-BERT over sentence classification tasks characterized by different training conditions, i.e., number of examples and number of categories. We report measures of our approach to support the development of deep learning models when exposed to few labeled examples over the following tasks: Topic Classification over the 20 News Group (20N) dataset (Lang, 1995), Question Classification (QC) on the UIUC dataset (Li and Roth, 2006), Sentiment Analysis over the SST-5 dataset (Socher et al., 2013). We

[2]From a computational perspective, the additional cost of $\mathcal{G}$ is negligible in terms of network parameters: it is an MLP which takes in input random vectors of 100 dimensions and produces in output vectors in the same 768-dimensional space of BERT. In other words, it is characterized by about 100 thousand parameters that are much less than in BERT base, i.e., 110 million parameters.



(a) 20N (b) QC Coarse Grained (c) QC Fine Grained

(d) SST-5 (e) MNLI Matched (f) MNLI Mismatched

Figure 2: Learning curves for the six tasks. We run all the models for 3 epochs except for 20N (15 epochs). The sequence length we used is: 64 for QC coarse, QC fine, and SST-5; 128 for both MNLI settings; 256 for 20N. Learning rate was set for all to 2e-5, except for 20N (5e-6).

will also report the performances over a sentence pair task, i.e., over the MNLI dataset (Williams et al., 2018). For each task, we report the performances with the metric commonly used for that specific dataset, i.e., accuracy for SST-5 and QC, while F1 is used for 20N and MNLI datasets. As a comparison, we report the performances of the BERT-base model fine-tuned as described in (Devlin et al., 2019) on the available training material. We used BERT-base as the starting point also for the training of our approach. GAN-BERT is implemented in Tensorflow by extending the original BERT implementation[3].

In more detail, $\mathcal{G}$ is implemented as an MLP with one hidden layer activated by a leaky-relu function. $\mathcal{G}$ inputs consist of noise vectors drawn from a normal distribution $N(0, 1)$. The noise vectors pass through the MLP and finally result in 768-dimensional vectors, that are used as fake examples in our architecture. $\mathcal{D}$ is, also, an MLP with one hidden layer activated by a leaky-relu function followed by a softmax layer for the final prediction. For both $\mathcal{G}$ and $\mathcal{D}$ we used dropout=0.1 after the hidden layer. We repeated the training of each model with an increasing set of annotated material (L), starting by sampling only 0.01% or 1% of the training set, in order to measure the performances starting with very few labeled examples (about 50-70 instances). GAN-BERT is also provided with a set of unlabeled examples U coming from the unused annotated material for each training set sample ($|U| = 100|L|$, when available). We replicated the labeled examples of a factor $log(|U|/|L|)$: this guarantees the presence of some labeled instances in each batch to avoid divergences due to the unsupervised component of the adversarial training. All the reported results are averaged over 5 different shuffles of the training material.

The 20N classification results are shown in figure 2a. The training and testing datasets are made of $11,314$ and $7,531$ documents classified in 20 categories[4], respectively. The plot shows F1 scores of the models: when 1% of data is used (i.e., about 110 examples) BERT almost diverges while GAN-BERT achieves more than 40% of F1. This trend is confirmed until 40% of labeled documents are used (i.e., about $5,000$ examples).

In the QC task we observe similar outcomes. The training dataset is made of about $5,400$ question. In the coarse-grained setting (figure 2b) 6 classes are involved; in the fine-grained scenario (figure 2c) the number of classes is 50. In both cases, BERT diverges when only 1% of labeled questions are used, i.e., about 50 questions. It starts to com-

[3] https://github.com/google-research/bert

[4] We used the train/test split available within scikit-learn.

pensate when using about 20% of the data in the coarse setting (about 1,000 labeled examples). In the fine-grained scenario, our approach is performing better until 50% of the labeled examples. It seems that, when a large number of categories is involved, i.e., the classification task is more complex, the semi-supervised setting is even more beneficial.

The results are confirmed in sentiment analysis over the SST-5 dataset (figure 2d), i.e., sentence classification involving 5 polarity categories. Also in this setting, we observe that GAN-BERT is beneficial when few examples are available. This is demonstrated by the difference in accuracy at 1% of the data (about 85 labeled examples), where BERT accuracy is 22.2% while GAN-BERT reaches 30.4% in accuracy. This trend is confirmed until about 20% of labeled examples (about 1,700), where BERT achieves comparable results.

Finally, we report the performances on Natural Language Inference on the MNLI dataset. We observe (in figures 2e and 2f) a systematic improvement starting from 0.01% labeled examples (about 40 instances): GAN-BERT provides about $6-10$ additional points in F1 with respect to BERT (18.09% vs. 29.19% and 18.01% vs. 31.64%, for mismatched and matched settings, respectively). This trend is confirmed until 0.5% of annotated material (about 2,000 annotated examples): GAN-BERT reaches 62.67% and 60.45% while BERT reaches 48.35% and 42.41%, for mismatched and matched, respectively. Using more annotated data results in very similar performances with a slight advantage in using GAN-BERT. Even if acquiring unlabeled examples for sentence pairs is not trivial, these results give a hint about the potential benefits on similar tasks (e.g., question-answer classification).

5 Conclusion

In this paper, we extended the limits of Transformed-based architectures (i.e., BERT) in poor training conditions. Experiments confirm that fine-tuning such architectures with few labeled examples lead to unstable models whose performances are not acceptable. We suggest here to adopt adversarial training to enable semi-supervised learning Transformer-based architectures. The evaluations show that the proposed variant of BERT, namely GAN-BERT, systematically improves the robustness of such architectures, while not introducing additional costs to the inference. In fact, the generator network is only used in training, while at inference time only the discriminator is necessary.

This first investigation paves the way to several extensions including adopting other architectures, such as GPT-2 (Radford et al., 2019) or DistilBERT (Sanh et al., 2019) or other tasks, e.g., Sequence Labeling or Question Answering. Moreover, we will investigate the potential impact of the adversarial training directly in the BERT pre-training. From a linguistic perspective, it is worth investigating what the generator encodes in the produced representations.

Acknowledgments

We would like to thank Carlo Gaibisso, Bruno Luigi Martino and Francis Farrelly of the Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti" (IASI) for supporting the early experimentations through access to dedicated computing resources made available by the Artificial Intelligence & High-Performance Computing laboratory.

References

Paolo Annesi, Danilo Croce, and Roberto Basili. 2014. Semantic compositionality in tree kernels. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 1029–1038. ACM.

Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. 2010. *Semi-Supervised Learning*, 1st edition. The MIT Press.

Danilo Croce, Giuseppe Castellucci, and Roberto Basili. 2019. Kernel-based generative adversarial networks for weakly supervised learning. In *AI*IA 2019 – Advances in Artificial Intelligence*, pages 336–347, Cham. Springer International Publishing.

Danilo Croce, Simone Filice, Giuseppe Castellucci, and Roberto Basili. 2017. Deep learning in semantic kernel spaces. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 345–354. Association for Computational Linguistics.

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

Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *J. Artif. Int. Res.*, 57(1):345–420.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2741–2749.

Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.

Xin Li and Dan Roth. 2006. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(3):229–249.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. 2016. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *International Conference on Acoustics, Speech and Signal Processing*, pages 5149–5152.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jason Weston, Frédéric Ratle, and Ronan Collobert. 2008. Deep learning via semi-supervised embedding. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1168–1175, New York, NY, USA. ACM.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 40–48. JMLR.org.