CERT: Contrastive Self-supervised Learning for Language Understanding

Hongchao Fang[†] COLEFANG.HONGCHAO@GMAIL.COM
UC San Diego

Pengtao Xie PENGTAOXIE2008@GMAIL.COM
UC San Diego

Abstract

Pretrained language models such as BERT, GPT have shown great effectiveness in language understanding. The auxiliary predictive tasks in existing pretraining approaches are mostly defined on tokens, thus may not be able to capture sentence-level semantics very well. To address this issue, we propose CERT: Contrastive self-supervised Encoder Representations from Transformers, which pretrains language representation models using contrastive self-supervised learning at the sentence level. CERT creates augmentations of original sentences using back-translation. Then it finetunes a pretrained language encoder (e.g., BERT) by predicting whether two augmented sentences originate from the same sentence. CERT is simple to use and can be flexibly plugged into any pretraining-finetuning NLP pipeline. We evaluate CERT on three language understanding tasks: CoLA, RTE, and QNLI. CERT outperforms BERT significantly.

1. Introduction

Large-scale pretrained language representation models such as BERT (Devlin et al., 2018), GPT (Radford et al., a), BART (Lewis et al., 2019), etc. have achieved dominating performance in various natural language processing tasks, such as text generation, reading comprehension, text classification, etc. The architectures of these models are mostly based on Transformer (Vaswani et al., 2017), which uses self-attention to capture long-range dependency between tokens. The Transformer encoder or decoder is pretrained on large-scale text corpus by solving self-supervised tasks, such as predicting masked tokens (Devlin et al., 2018), generating future tokens (Radford et al., a), denoising corrupted tokens (Lewis et al., 2019), etc. In these works, the targets to be predicted are mostly at the word level. As a result, the global semantics at the sentence level may not be sufficiently captured.

To address this issue, we propose CERT: Contrastive self-supervised Encoder Representations from Transformers, which uses contrastive self-supervised learning (CSSL) (He et al., 2019; Chen et al., 2020a) to learn sentence-level representations. Recently, CSSL has shown promising results in learning visual representations in an unsupervised way (He et al., 2019; Chen et al., 2020a). The key idea of CSSL is: create augments of original examples, then learn representations by predicting whether two augments are from the same original data example or not. CERT creates augments of sentences using back-translation (Edunov et al.,

. †The work was done during internship at UCSD.

2018), then finetunes a pretrained language representation model (e.g., BERT, BART) by predicting whether two augments are from the same original sentence or not. Different from existing pretraining methods where the prediction tasks are defined on tokens, CERT defines the prediction task on sentences, which can presumably better capture global semantics at the sentence level.

CERT uses back-translation (Edunov et al., 2018) to perform sentence augmentation. Given a sentence x in source language S, we use an S-to-T translation model to translate x into a sentence y in target language T. Then we use a T-to-S translation model to translate y into x' in the source language. x' is regarded as an augment of x. Translation models for different target languages are used to create different augments of a source sentence. Given these augmented sentences, a Momentum Contrast (MoCo) (He et al., 2019) method is used to perform CSSL. MoCo maintains a queue of augmented sentences (called keys) which are encoded using a pretrained text-encoder (e.g., BERT) with momentum updates. Given an augmented sentence (called query), a similarity score is calculated between the BERT (or any other pretrained text-encoder) encoding of the query and each key in the queue. The query and a key is labeled as a positive pair if they are augments of the same original sentence and as a negative pair if otherwise. These binary labels and similarity scores are used to define contrastive losses (Hadsell et al., 2006). The weights of the pretrained text encoder are further finetuned by minimizing the contrastive losses. To apply the pretrained CERT model on a downstream task, we finetune the CERT weights using input data and labels from the downstream task. CERT is a flexible module that can be integrated with any pretrained language representation models, such as BERT, BART, ERNIE 2.0 (Sun et al., 2019), T5 (Raffel et al., 2019), etc.

We evaluate CERT on three language understanding tasks in the GLUE benchmark (Wang et al., 2018): linguistic acceptability on the CoLA dataset (Warstadt et al., 2019), textual entailment on the RTE dataset (Dagan et al., 2005) and question-answering natural language inference (QNLI) (Rajpurkar et al., 2016). On these tasks, CERT outperforms BERT significantly, which demonstrates the effectiveness of contrasive self-supervised learning for language representation.

The major contributions of this paper are as follows:

- We propose CERT, a new language representation pretraining method based on contrastive self-supervised learning. The predictive tasks of CERT are defined at the sentence level, thus presumably can better capture sentence-level semantics.
- We perform evaluation of CERT on three tasks – CoLA, RTE, and QNLI. CERT performs significantly better than BERT on these tasks.

The rest of the papers are organized as follows. Section 2 and 3 present the methods and experiments. Section 4 reviews related works and Section 5 concludes the paper.

2. Pre-training of Transformers for Language Understanding

Among the recent works for pretraining language representation models, most of them are based on the Transformer (Vaswani et al., 2017) architecture. For example, BERT pretrains Transformer encoder. GPT pretrains Transformer decoder. BART pretrains Transformer encoder and decoder jointly.

2.1. Transformer

Transformer (Vaswani et al., 2017) is an encode-decoder architecture for sequence-to-sequence (seq2seq) modeling (Sutskever et al., 2014). Different from seq2seq models (Sutskever et al., 2014) that are based on recurrent neural networks (e.g., LSTM (Hochreiter and Schmidhuber, 1997), GRU (Chung et al., 2014)) which model a sequence of tokens via a recurrent manner and hence is computationally inefficient. Transformer eschews recurrent computation and instead uses self-attention which not only can capture the dependency between tokens but also is amenable for parallel computation with high efficiency. Self-attention calculates the correlation among every pair of tokens and uses these correlation scores to create "attentive" representations by taking weighted summation of tokens' embeddings. Transformer is composed of building blocks, each consisting of a self-attention layer and a position-wise feed-forward layer. Residual connection (He et al., 2016) is applied around each of the two sub-layers, followed by layer normalization (Ba et al., 2016). Given the input sequence, an encoder, which is a stack of such building blocks, is applied to obtain a representation for each token. Then the decoder takes these representations as inputs and decodes the sequence of output tokens. To decode the i-th token, the decoder first uses self-attention to encode the already decoded sequence $y_1, \cdots, y_{i-1}$, then performs input-output attention between the encodings of $y_1, \cdots, y_{i-1}$ and those of the input sequence. The "attentive" representations are then fed into a feed-forward layer. The three steps are repeated for multiple times. Finally, the representation is fed into a linear layer to predict the next token. The weight parameters in Transformer is learned by maximizing the conditional likelihood of output sequences conditioned on the corresponding input sequences.

2.2. BERT

BERT (Devlin et al., 2018) aims to learn a Transformer encoder for representing texts. BERT's model architecture is a multi-layer bidirectional Transformer encoder. In BERT, the Transformer uses bidirectional self-attention. To train the encoder, BERT masks some percentage of the input tokens at random, and then predicts those masked tokens by feeding the final hidden vectors (produced by the encoder) corresponding to the mask tokens into an output softmax over the vocabulary. To apply the pretrained BERT to a downstream task such as sentence classification, one can add an additional layer on top of the BERT architecture and train this newly-added layer using the labeled data in the target task.

3. Contrastive Self-supervised Learning

Self-supervised learning (SSL) (Wu et al., 2018; He et al., 2019; Chen et al., 2020b,a) is a learning paradigm which aims to capture the intrinsic patterns and properties of input data without using human-provided labels. The basic idea of SSL is to construct some auxiliary tasks solely based on the input data itself without using human-annotated labels and force the network to learn meaningful representations by performing the auxiliary tasks well.

The auxiliary tasks in SSL can be constructed using many different mechanisms. Recently, a contrastive mechanism (Hadsell et al., 2006) has gained increasing attention and demonstrated promising results in several studies (He et al., 2019; Chen et al., 2020b). The basic idea of contrastive SSL is: generate augmented examples of original data examples,

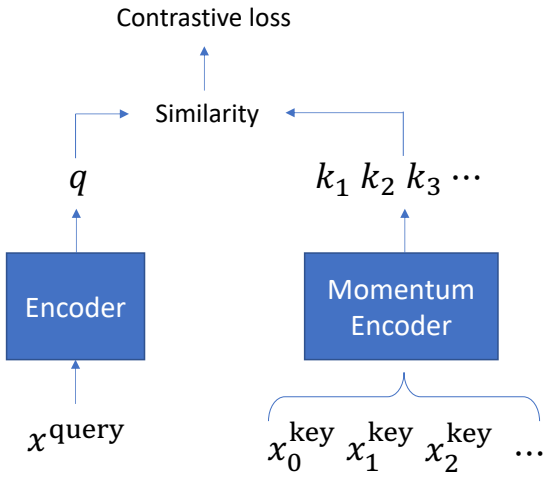

Figure 1: The keys in the queue are encoded using a momentum encoder. Given an augmented data example in the current minibatch (called query) and a key in the queue, they are considered as a positive pair if they originate from the same data example, and a negative pair if otherwise. A similarity score is calculated between the encoding of the query and the encoding of each key. Contrastive losses are defined on the similarity scores and binary labels.

create a predictive task where the goal is to predict whether two augmented examples are from the same original data example or not, and learn the representation network by solving this task.

Different methods have been proposed to implement contrastive SSL. In SimCLR (Chen et al., 2020a) designed for image data, given the input images, random data augmentation is applied to these images. If two augmented images are created from the same original image, they are labeled as being similar; otherwise, they are labeled as dissimilar. Then SimCLR learns a network to fit these similar/dissimilar binary labels. The network consists of two modules: a feature extraction module $f(\cdot)$ which extracts the latent representation $\mathbf{h} = f(\mathbf{x})$ of an image $\mathbf{x}$ and a multi-layer perceptron $g(\cdot)$ which takes $\mathbf{h}$ as input and generates another latent representation $\mathbf{z} = g(\mathbf{h})$ used for predicting whether two images are similar. Given a similar pair $(\mathbf{x}_i, \mathbf{x}_j)$ and a set of images $\{\mathbf{x}_k\}$ that are dissimilar from $\mathbf{x}_i$, a contrastive loss can be defined as follows:

$$-\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j))/\tau)}{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j))/\tau) + \sum_k \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k))/\tau)} \tag{1}$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity between two vectors and τ is a temperature parameter. SimCLR learns the network weights by minimizing losses of this kind. After training, the feature extraction sub-network is used for downstream tasks and $g(\cdot)$ is discarded.

While SimCLR is easy to implement, it requires a large minibatch size to yield high performance, which is computationally prohibitive. MoCo (Chen et al., 2020a) addresses

this problem by using a queue that is independent of minibatch size. This queue contains a dynamic set of augmented data examples (called keys). In each iteration, the latest minibatch of examples are added into the queue; meanwhile, the oldest minibatch is removed from the queue. In this way, the queue is decoupled from minibatch size. Figure 1 shows the architecture of MoCo. The keys are encoded using a momentum encoder. Given an augmented data example in the current minibatch (called query) and a key in the queue, they are considered as a positive pair if they originate from the same image, and a negative pair if otherwise. A similarity score is calculated between the encoding of the query and the encoding of each key. Contrastive losses are defined on the similarity scores and binary labels.

4. CERT

CERT takes a pretrained language representation model (e.g., BERT) and finetunes it using contrastive self-supervised learning on the input data of the target task.

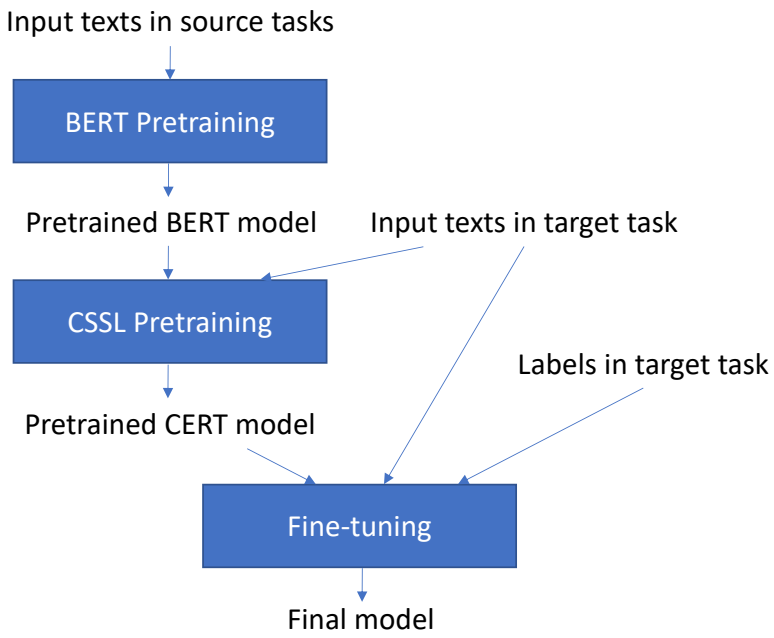

Figure 2: The workflow of CERT. Given the large-scale input texts (without labels) from source tasks, a BERT model is first pretrained on these texts. Then we continue to train this pretrained BERT model using CSSL on the input texts (without labels) from the target task. We refer to this model as pretrained CERT model. Then we finetune the CERT model using the input texts and their associated labels in the target task and get the final model that performs the target task.

Figure 2 shows the workflow of CERT. For the ease of presentation, we use BERT as a running example of the pretrained language representation model. But note that CERT

can be used on top of other pretrained language representation models such as XLNet, T5, etc. as well and is not specific to BERT. Given the large-scale input texts (without labels) from source tasks, a BERT model is first pretrained on these texts. Then we continue to train this pretrained BERT model using CSSL on the input texts (without labels) from the target task. We refer to this model as pretrained CERT model. Then we finetune the CERT model using the input texts and their associated labels in the target task and get the final model that performs the target task. In CSSL training, CERT augments the original sentences in the target task using back-translation. Two augmented sentences are considered as a positive pair if they are created from the same original sentence and a negative pair if otherwise. The augmented sentences are encoded with BERT and a similarity score is calculated on the BERT encodings of a pair of sentences. Contrastive losses are defined on the binary labels and similarity scores. Then the pretrained BERT encoder is further finetuned by minimizing the contrastive losses. We use MoCo to implement CSSL to avoid using large minibatches which are computationally heavy.

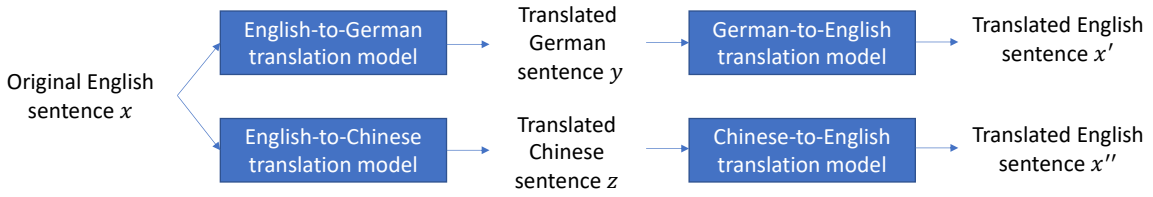

Figure 3: The workflow of data augmentation based on back translation.

Data Augmentation For each input sentence x in the target task, we augment it using back-translation (Edunov et al., 2018). Without loss of generality, we assume the language in the target task is English. We use an English-to-German machine translation (MT) model to translate x to y. Then we use a German-to-English translation model to translate y to x'. Then x' is regarded as an augmented sentence of x. Similarly, we use an English-to-Chinese MT model and a Chinese-to-English MT model to obtain another augmented sentence x''. We use the machine translation models developed in (Britz et al., 2017) for back-translation.

CSSL Pretraining We use MoCo (He et al., 2019) to implement CSSL. Given two augmented sentences, if they originate from the same original sentence, they are labeled as a positive pair; if they are from different sentences, they are labeled as a negative pair. We use a queue to maintain a set of augmented sentences $\{k_i\}_{i=1}^K$ (called keys). Given an augmented sentence q (called query) in the currently sampled minibatch, it is compared against each key in the queue. The query is encoded with a pretrained BERT model $f_q(\cdot;\theta_q)$ where θ_q denotes the weights. Each key is encoded with a pretrained BERT $f_k(\cdot;\theta_k)$, where the weights θ_k are updated with momentum: $\theta_k \leftarrow m\theta_k + (1-m)\theta_q$ (m is the momentum coefficient). Without loss of generality, we assume there is a single key k_+ in the queue that forms a positive pair with q. A contrastive loss can be defined as:

$$-\log \frac{\exp(f_q(q;\theta_q)\cdot f_k(k_+;\theta_k)/\tau)}{\sum_{i=1}^K \exp(f_q(q;\theta_q)\cdot f_k(k_i;\theta_k)/\tau)} \tag{2}$$

where τ is a temperature parameter. The weights of the BERT encoder are finetuned by minimizing losses of such kind. Note that in this step, only the input sentences of the target task are used. The labels of these sentences are not touched. To apply the pretrained CERT model to a downstream task, we further finetune its weights on both the input sentences and their labels in the target task.

5. Experiments

In this section, we evaluate CERT on three language understanding tasks in the GLUE benchmark (Wang et al., 2018): linguistic acceptability (Warstadt et al., 2019), textual entailment (Dagan et al., 2005), and question-answering language inference (Rajpurkar et al., 2016).

5.1. Tasks and Datasets

CoLA The Corpus of Linguistic Acceptability (Warstadt et al., 2019) is comprised of English acceptability judgments extracted from articles and books about linguistic theory. The task is to judge whether a sequence of words is a valid English sentence with correct grammar. There are 8551 training examples (each consisting of a sequence of words and a binary label regarding whether this word sequence is a grammatically-correct sentence), 1043 validation examples, and 1064 test examples. The labels in the test set are not released. We use the validation set for performance evaluation. Matthews correlation coefficient (Matthews, 1975) is used as the evaluation metric where the range is $[-1, 1]$. The higher, the better. The CSSL training in CERT is on the 8551 sentences in the training set.

RTE The Recognizing Textual Entailment (RTE) task is defined as follows: given two sentences, judge whether one sentence can entail another. Entailment means the truth of one sentence follows from the other sentence. The datasets are collected from several annual textual entailment challenges, including RTE1 (Dagan et al., 2005), RTE2 (Haim et al., 2006), RTE3 (Giampiccolo et al., 2007), and RTE5 (Bentivogli et al.). The sentences are from news and Wikipedia text. All datasets are converted to a two-class split: for three-class datasets, neutral and contradiction are collapsed into not entailment. In the training set, there are 2491 pairs of sentences. Each pair is labeled as either "entailment" or "not entailment". The validation set has 278 labeled pairs and the test set has 2985 unlabeled pairs. We use accuracy as the evaluation metric. The performance is measured on the validation set.

QNLI The Stanford Question Answering Dataset (Rajpurkar et al., 2016) is an open-ended question-answering dataset. Given a Wikipedia paragraph and a question, the task is to identify a text span from the paragraph as the answer to this question. From this dataset, Wang et al. (2018) creates a question-answering natural language inference (QNLI) task: given a pair of sentences (q, p) where q is a question, judge whether p contains the answer of q. The dataset is split into a training set, a validation set, and a test, containing 104742, 5462, and 5462 pairs respectively. Each pair in the training and validation set has a binary label regarding whether the second sentence contains the answer of the question (the first sentence). Since labels in the test set are not released, we use the validation set to

evaluate the performance. The evaluation metric is accuracy. The CSSL training in CERT is on the sentences in the training set.

5.2. Experimental Settings

In MoCo, the size of the queue was set to 96606. The coefficient of MoCo momentum of updating key encoder was set to 0.999. The temperature parameter in contrastive loss was set to 0.07. Multi-layer perceptron head was used. For MoCo training, a stochastic gradient descent solver with momentum was used. The minibatch size was set to 16. The initial learning rate was set to 4e-5. The learning rate was adjusted using cosine scheduling. The number of epochs was set to 100. Weight decay was used with a coefficient of 1e-5.

For CoLA finetuning, the maximum sequence length was set to 128. Minibatch size was set to 16. Learning rate was set to 3e-5. The number of training epochs was set to 20. For RTE finetuning, the maximum sequence length was set to 128. Minibatch size was set to 16. Learning rate was set to 2e-5. The number of training epochs was set to 5. For QNLI finetuning, the maximum sequence length was set to 128. Minibatch size was set to 16. Learning rate was set to 2e-5. The number of training epochs was set to 3.

5.3. Results

Table 1 shows Matthews correlation on the CoLA validation set. Our CERT method achieves a score of 62.6, which significantly improves upon BERT, which achieves a score of 60.6. This demonstrates the effectiveness of contrastive self-supervised learning in better capturing global semantics at the sentence level. For the convenience of the readers, we also show the results of state-of-the-art pretraining methods including XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019), ERNIE 2.0 (Sun et al., 2019), and ALBERT (Lan et al., 2019). CERT achieves a performance close to XLNet, with a much lower consumption of computing resources and on a much smaller text corpus. XLNet, RoBERTa, ERNIE 2.0, and ALBERT are trained on hundreds of to thousands of GPU machines for several days, while CERT is trained using a single GPU for a dozen of hours. In addition, these models are trained on tens of or hundreds of gigabytes of texts while CERT is only trained on about 400 KB of texts. CERT can be built on top of these pretrained models as well, which will be left for future study.

	Matthews correlation (%)
BERT	60.6
CERT (ours)	62.6
XLNet	63.6
ERNIE 2.0	65.4
RoBERTa	68.0
ALBERT	71.4

Table 1: Matthews correlation on CoLA validation set.

Table 2 shows the accuracy on the RTE validation set. CERT achieves an accuracy of 74.0%, which is largely better than the 70.4% accuracy achieved by BERT. This further demonstrates the effectiveness of CSSL pretraining. Table 3 shows accuracy on the QNLI

validation set. CERT performs better than BERT, though the improvement is not as significant as that in CoLA and RTE. One possible reason is that the QNLI training data is much larger than CoLA and RTE, and therefore is more robust to overfitting, which makes the necessity of CSSL pretraining less prominent.

	Accuracy (%)
BERT	70.4
CERT (ours)	74.0
XLNet	83.8
ERNIE 2.0	85.2
RoBERTa	86.6
ALBERT	89.2

Table 2: Accuracy on RTE validation set.

	Accuracy (%)
BERT	92.1*
CERT (ours)	92.5
XLNet	93.9
ERNIE 2.0	94.3
RoBERTa	94.7
ALBERT	95.3

Table 3: Accuracy on QNLI validation set. (*)The reported performance in (Lan et al., 2019) is 92.3, which we were not able to reproduce.

6. Related Works

6.1. Pretraining for learning language representation

Recently, pretraining on large-scale text corpus for language representation learning has achieved substantial success. The GPT model (Radford et al., a) is a language model (LM) based on Transformer. Different from Transformer which defines a conditional probability on an output sequence given an input sequence, GPT defines a marginal probability on a single sequence. In GPT, the conditional probability of the next token given the historical sequence is defined using the Transformer decoder. The weight parameters are learned by maximizing the likelihood on the sequence of tokens. GPT-2 (Radford et al., b) is an extension of GPT, which modifies GPT by moving layer normalization to the input of each sub-block and adding an additional layer normalization after the final self-attention block. Byte pair encoding (BPE) (Sennrich et al., 2015) is used to represent the input sequence of tokens. BERT-GPT (Wu et al., 2019) is a model used for sequence-to-sequence modeling where pretrained BERT is used to encode the input text and GPT is used to generate the output text. In BERT-GPT, the pretraining of the BERT encoder and the GPT decoder

is conducted separately, which may lead to inferior performance. Auto-Regressive Transformers (BART) (Lewis et al., 2019) has a similar architecture as BERT-GPT, but trains the BERT encoder and GPT decoder jointly. To pretrain the BART weights, the input text is corrupted randomly, such as token masking, token deletion, text infilling, etc., then the network is learned to reconstruct the original text. ALBERT (Lan et al., 2019) uses parameter-reduction methods to reduce the memory consumption and increase the training speed of BERT. It also introduces a self-supervised loss which models inter-sentence coherence. RoBERTa (Liu et al., 2019) is a replication study of BERT pretraining. It shows that the performance of BERT can be significantly improved by carefully tuning the training process, such as (1) training the model longer, with bigger batches, over more data; (2) removing the next sentence prediction objective; (3) training on longer sequences, etc. XLNet (Yang et al., 2019) learns bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order and uses a generalized autoregressive pretraining mechanism to overcome the pretrain-finetune discrepancy of BERT. T5 (Raffel et al., 2019) compared pre-training objectives, architectures, unlabeled datasets, transfer approaches on a wide range of language understanding tasks and proposed a unified framework that casts these tasks as a text-to-text task. The unified model was trained on a large Colossal Clean Crawled Corpus, which was then transferred to diverse downstream tasks. ERNIE 2.0 (Sun et al., 2019) proposed a continual pre-training framework which builds and learns incrementally pre-training tasks through constant multi-task learning, to capture the lexical, syntactic and semantic information from training corpora.

6.2. Self-supervised learning

Self-supervised learning (SSL) aims to learn meaningful representations of input data without using human annotations. It creates auxiliary tasks solely using the input data and forces deep networks to learn highly-effective latent features by solving these auxiliary tasks. Various strategies have been proposed to construct auxiliary tasks, based on temporal correspondence (Li et al., 2019; Wang et al., 2019a), cross-modal consistency (Wang et al., 2019b), etc. Examples of auxiliary tasks include rotation prediction (Gidaris et al., 2018), image inpainting (Pathak et al., 2016), automatic colorization (Zhang et al., 2016), context prediction (Nathan Mundhenk et al., 2018), etc. Some recent works studied self-supervised representation learning based on instance discrimination (Wu et al., 2018) with contrastive learning. Oord et al. proposed contrastive predictive coding (CPC) to extract useful representations from high-dimensional data (Oord et al., 2018). Bachman et al. proposed a self-supervised representation learning approach based on maximizing mutual information between features extracted from multiple views of a shared context (Bachman et al., 2019). Most recently, Chen et al. presented a simple framework for contrastive learning (SimCLR) (Chen et al., 2020a) with larger batch sizes and extensive data augmentation (Lee et al., 2019), which achieved results that are comparable with supervised learning. Momentum Contrast (MoCo) (He et al., 2019; Chen et al., 2020b) expanded the idea of contrastive learning with an additional dictionary and a momentum encoder. He et al. (2020) proposed an Self-Trans approach which applies contrastive self-supervised learning on top of networks pretrained by transfer learning.

7. Conclusions

In this work, we propose CERT, a pretraining approach for language representation learning. CERT takes a pretrained language representation model such as BERT and continues to train it using contrastive self-supervised learning on the input texts of the target task. It uses back-translation to generate augmented sentences for each original sentence in the target data, and trains the network by predicting whether two augments are from the same original sentence or not. Then the pretrained CERT model is finetuned on the input texts and their labels in the target task. On three natural lanuage understanding tasks, CERT outperforms BERT significantly, demonstrating the effectiveness of contrastive self-supervised learning for language representation.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*, 2017.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020a.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer, 2005.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*, 2018.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics, 2007.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Xuehai He, Xingyi Yang, Shanghang Zhang, Jinyu Zhao, Yichen Zhang, Eric Xing, and Pengtao Xie. Sample-efficient deep learning for covid-19 diagnosis based on ct scans. *medRxiv*, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Rethinking data augmentation: Self-supervision and self-distillation. *arXiv preprint arXiv:1910.05872*, 2019.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *Advances in Neural Information Processing Systems*, pages 317–327, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9339–9348, 2018.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. a.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. b.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*, 2019.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019a.

Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019b.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Qingyang Wu, Lei Li, Hao Zhou, Ying Zeng, and Zhou Yu. Importance-aware learning for neural headline editing. *arXiv preprint arXiv:1912.01114*, 2019.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.