Self-training Improves Pre-training for Natural Language Understanding

Jingfei Du[†*] **Edouard Grave**[†] **Beliz Gunel**[‡] **Vishrav Chaudhary**[†]

Onur Celebi[†] **Michael Auli**[†] **Veselin Stoyanov**[†] **Alexis Conneau**[†*]

{jingfeidu,egrave,vishrav,celebi,michaelauli,ves,aconneau}@fb.com, bgunel@stanford.edu

[†]Facebook AI, [‡]Stanford University

Abstract

Unsupervised pre-training has led to much recent progress in natural language understanding. In this paper, we study self-training as another way to leverage unlabeled data through semi-supervised learning. To obtain additional data for a specific task, we introduce SentAugment, a data augmentation method which computes task-specific query embeddings from labeled data to retrieve sentences from a bank of billions of unlabeled sentences crawled from the web. Unlike previous semi-supervised methods, our approach does not require in-domain unlabeled data and is therefore more generally applicable. Experiments show that self-training is complementary to strong RoBERTa baselines on a variety of tasks. Our augmentation approach leads to scalable and effective self-training with improvements of up to 2.6% on standard text classification benchmarks. Finally, we also show strong gains on knowledge-distillation and few-shot learning.

1 Introduction

Self-training is a semi-supervised method which uses a teacher model, trained using labeled data, to create synthetic labels for unlabeled examples (Scudder, 1965; Yarowsky, 1995). These synthetic labels are then used to train a student model. This approach is called self-training when the student model has a similar or higher capacity than the teacher, and knowledge distillation (Hinton et al., 2015) when the student model is smaller than the teacher. Self-training has been successfully applied to a variety of tasks, including image recognition (Yalniz et al., 2019; Xie et al., 2020; Zoph et al., 2020), automatic speech recognition (Synnaeve et al., 2019; Kahn et al., 2020; Park et al., 2020), sequence generation (He et al., 2019), and parsing (McClosky et al., 2006).

An alternative semi-supervised technique is pre-training (Dai and Le, 2015; Radford et al., 2018; Howard and Ruder, 2018; Devlin et al., 2018), which has led to large improvements for natural language understanding compared to purely supervised learning. In that case, models are first trained on an auxiliary task, such as language modeling, followed by fine-tuning on the task of interest.

A natural question is the following: *do pre-training and self-training capture the same information, or are they complementary?* Recently, Zoph et al. (2020) studied this question in the context of image recognition, showing that self-training was helpful, even in addition to pre-training. However, their study mostly considers supervised pre-training, in which models were trained on ImageNet classification. Moreover, in cases where large amounts of supervised data were available for the downstream task, pre-training was not helpful, even without self-training. This is in contrast to natural language understanding for which language modeling pre-training is a very strong baseline that leads to large improvements for all the tasks we consider.

An important ingredient for self-training, and semi-supervised learning in general, is the unannotated data and the fact that it comes from the same domain as the downstream task. Existing work, such as UDA (Xie et al., 2019), self-training (He et al., 2019; Xie et al., 2020) and back-translation for machine translation (Bojar and Tamchyna, 2011; Sennrich et al., 2015; Edunov et al., 2018), assumes the existence of unannotated data in the same domain as the downstream task. This assumption limits the broad application of such semi-supervised methods, in particular in the case of low-resource downstream tasks. A second important question is thus: *how can we obtain large amounts of unannotated data from specific domains?*

In this paper, we propose a data augmentation

[*]Equal contribution.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5408–5418
June 6–11, 2021.

method, SentAugment, to build datasets of "in-domain" data for a given task from data crawled on the web. Web data covers many domains, and is available in large quantities. We use a large bank of web documents and construct sentence embeddings (Kiros et al., 2015; Wieting et al., 2016; Conneau et al., 2017; Artetxe and Schwenk, 2019; Cer et al., 2018; Arora et al., 2017) that allow us to retrieve domain-specific unannotated sentences, which are similar to the existing training set of the downstream tasks. Our sentence embedding model is optimized for similarity search, trained with a triplet loss on ground-truth paraphrases, parallel sentences as well as as hard negatives (Wieting et al., 2016; Wieting and Gimpel, 2017). We train a teacher model using the labeled task data and then further use it to synthetically label the retrieved sentences, and train the final model based on this synthetic dataset. Experiments show that SentAugment is effective for self-training, knowledge distillation and few-shot learning. The approach is generally applicable to new problems, leading to improvements on a variety of domains and tasks such as hate-speech and movie review classification over a strong RoBERTa (Devlin et al., 2018; Liu et al., 2019) baseline. To the best of our knowledge, this is the first study showing that self-training is complementary to a strong pre-training baseline for natural language understanding. Specifically, we make the following contributions:

- We introduce SentAugment, a data augmentation approach for semi-supervised learning that retrieves task-specific in-domain data from a large bank of web sentences.
- We show that self-training improves upon unsupervised pretraining: we improve RoBERTa-Large by 1.2% accuracy on average on six standard classification benchmarks.
- We show that self-training improves accuracy by 3.5% on average for few-shot learning.
- For knowledge-distillation, our approach improves the distilled RoBERTa-Large by 2.9% accuracy on average, reducing the gap between the teacher and the student model.
- We release code and models for researchers to build on top of our work.[1]

[1] https://github.com/facebookresearch/SentAugment

2 Approach

Our SentAugment approach retrieves task-specific in-domain unsupervised data from a large bank of sentences which is used for self-training, where the teacher model - a RoBERTa-Large model finetuned on the downstream task - synthetically labels it. The synthetic labeled data is finally used to train the output student model (see Figure 1). We give more details on our approach in what follows.

2.1 SentAugment: data augmentation for semi-supervised learning

Whereas most semi-supervised approaches rely on in-domain unlabeled data, we are constructing similar datasets on the fly from the large bank of unannotated text. In what follows, we describe our data retrieval strategy for augmentation.

Large-scale sentence bank. Our approach relies on a large-scale corpus of unsupervised sentences, derived from data crawled on the web (Wenzek et al., 2019). Because of its scale and diversity, our sentence bank contains data from various domains and with different styles, allowing to retrieve relevant data for many downstream tasks. We embed each sentence using a universal paraphrastic sentence encoder (Wieting et al., 2016; Arora et al., 2017; Ethayarajh, 2018a), a model which was trained to output similar representations for sentences of similar meaning. This sentence embedding space does not depend on the downstream tasks, and will be used to retrieve subsets of the sentence bank which are relevant to particular tasks. For sentence encoders, we consider word2vec embeddings (Mikolov et al., 2013, 2018) and uSIF (Ethayarajh, 2018b). We also train our own English sentence encoder, a Transformer pretrained with masked language modeling and finetuned to maximize cosine similarity between similar sentences. Specifically, we use a triplet loss $\mathcal{L}(x, y) = max(0, \alpha - cos(x, y) + cos(x, y_c))$ where positive pairs (x, y) are either paraphrases or parallel sentences (Wieting et al., 2019a) and y_c are in-batch hard negatives (Wieting et al., 2016).

Downstream task embeddings. For each downstream task, we build embeddings that are representative of the task, using the same paraphrastic model. Then, we use these *task embeddings* as queries for retrieving similar sentences from the sentence bank, using cosine similarity in the embedding space. Specifically, we consider three ways

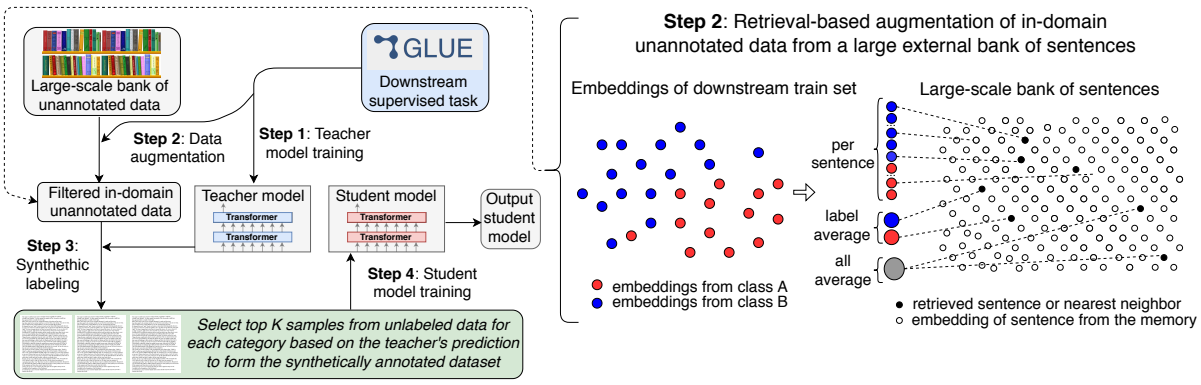

Figure 1: **The SentAugment approach.** The self-training procedure follows multiple steps; **Step 1**: A RoBERTa-Large model - the teacher - is finetuned on a downstream task using a cross-entropy loss, **Step 2**: Task-specific unannotated data is extracted from a large bank of sentences; This step uses task-specific query embeddings (produced by a paraphrastic sentence encoder) to select nearest neighbors from the bank. **Step 3**: This data is synthetically annotated using the teacher model; top K samples from each class are selected to form the final synthetic dataset; **Step 4**: A RoBERTa-Large model - the student - is finetuned on this dataset using KL-divergence. Our approach differs from previous work at Step 2, which we show is crucial for open-domain self-training.

for computing the task embeddings: *all-average*, where we obtain one embedding by averaging the sentence embeddings of all the samples from the training set of the downstream task ; *label-average*, where we construct one embedding per label, corresponding to the average of the sentence embeddings in the train set for each label ; *per-sentence*, where we keep one embedding for each sentence on the training set of the downstream task.

Unsupervised data retrieval. Using task-representative embeddings as queries, we retrieve a subset of our large sentence bank, corresponding to a few million sentences which we use as in-domain candidates for semi-supervised learning. Reducing the amount of unannotated data is an important step as synthetically annotating billions of sentences using a large Transformer does not scale. We perform additional filtering based on the confidence of our teacher model keeping only high-confident samples while maintaining the ratio of labels of the training set of the downstream task. For relatively small tasks, we use a threshold such that our augmented training set is approximately a hundred times bigger, and for datasets of medium size, only ten times bigger.

2.2 Semi-supervised learning for natural language understanding

We combine our data augmentation technique with self-training and knowledge distillation, two semi-supervised learning techniques that benefit from having relevant unannotated sentences.

Self-training. Following the steps in Figure 1, we first train a teacher model by fine-tuning a pre-trained RoBERTa-Large model on the target downstream task. We then use it to annotate the retrieved in-domain sentences. For each class, we select the sentences with the highest scores and prune the rest. We make sure the label ratio is maintained between the original downstream task training set and the augmented set by considering the probability of the classifier. As our student model, we then finetune a new RoBERTa-Large using KL-divergence on the synthetic data by considering the post-softmax class probabilities as labels.

Knowledge-distillation. We follow the same approach for knowledge-distillation, except we consider a student model that has an order of magnitude less parameters than the RoBERTa-Large teacher model. As for self-training, we pretrain the student and use continuous probabilities as synthetic labels. We exploit data augmentation by using in-domain unannotated sentences.

Few-shot learning. Semi-supervised learning techniques are adapted to settings where little supervised data is available. We simulate a few-shot learning environment by only considering a few samples per class, for several downstream tasks. We apply data augmentation and self-training in that context by augmenting the training set by two to three orders of magnitude more data and use a teacher model trained on only a few training samples to synthetically annotate data.

Dataset	task	domain	#train	#classes
SST-2	sentiment analysis	movie reviews	67349	2
SST-5	sentiment analysis	movie reviews	8544	5
CR	product classification	product reviews	2500	2
IMP	hate-speech classification	forum conversations	3947	2
TREC	question-type classification	short questions	5001	6
CoNLL	named entity recognition	news stories	11663	5

Table 1: Downstream tasks used for evaluation.

3 Experimental setup

Next, we give details on how we build the bank of sentences, what downstream tasks we use for evaluation and we describe our training procedure for semi-supervised learning.

3.1 Large-scale bank of sentences

As a large-scale external bank of unannotated sentences, we extract and filter text from CommonCrawl [2] (Wenzek et al., 2019). In particular, we apply a simple sentence segmenter to turn documents into sentences and perform deduplication. We refer to samples in this dataset as sentences although is also contains shorts spans of text that can be seen as short documents. We use three corpora, CC-100M with one hundred million sentences (2B words), CC-1B with one billion sentences (20B words) and CC-5B with five billion sentences (100B words), the first two being random subsets of the biggest one. When retrieving sentences, we remove those that overlap with sentences from the test set of the downstream task. CommonCrawl data contains a wide variety of domains and text styles which makes it a good general-purpose corpus. We release pointers to obtain a similar corpus.

3.2 Evaluation datasets

We evaluate our approach on the Stanford Sentiment Treebank (Socher et al., 2013) binary and fine-grained sentiment analysis datasets (SST-2 and SST-5), on product classification (CR) from (Hu and Liu, 2004), hate-speech comment classification[3] (IMP), question classification (TREC) from (Voorhees and Tice, 2000) and named entity recognition (CoNLL 2002) from (Sang and De Meulder, 2003). We provide details of each task including task, domain, size and number of classes in Table 1.

[2] www.github.com/facebookresearch/cc_net

[3] www.kaggle.com/c/detecting-insults-in-social-commentary/overview

3.3 Training details

Our sentence embeddings. We train our own SentAugment Sentence Encoder (SASE) by leveraging paraphrases from NLI entailment pairs (Williams et al., 2017), MRPC (Dolan and Brockett, 2005), Quora Question Pairs (QQP), round-trip translation (Wieting and Gimpel, 2017) and web paraphrases (Creutz et al., 2018), together with OpenSubtitles (Lison et al., 2019) and Europarl (Koehn, 2005) parallel data from English to French, Italian and Indonesian - language pairs that were shown to provide good paraphrastic sentence embeddings (Wieting et al., 2019a). We pretrain the model with a multilingual masked language modeling objective (Devlin et al., 2018; Conneau and Lample, 2019) in these 4 languages, with a sentence piece segmentation trained on a corpus with 3/4 of English data to give more importance to English, and the rest in other languages. We use a triplet loss to learn cosine sentence embedding similarity where the negative is selected to be the hardest in the batch. We evaluate our model on STS benchmarks (Agirre et al., 2012) and report results in Section 5 where we show our model outperforms previous approaches. We found that due to pretraining and being trained on longer sentences, our model is also more adapted to raw and long sentences from CommonCrawl. We also consider word2vec embeddings (Mikolov et al., 2013) and the uSIF approach (Ethayarajh, 2018b; Arora et al., 2017) as baselines in our experimental results.

Fine-tuning the student model. We use fairseq (Ott et al., 2019) and the open-source RoBERTa-Large model (Liu et al., 2019) as our pretrained Transformer baseline and perform finetuning on each downstream task. We use Adam, with learning-rate schedule 1e-5. We use batch-sizes of 16 and dropout rate 0.1. We fine-tune on synthetically annotated data using

Model	SST-2	SST-5	CR	IMP	TREC	NER	Avg
$\text{RoBERTa}_{\text{Large}}$	96.5	57.8	94.8	84.6	**97.8**	92.7	87.4
$\text{RoBERTa}_{\text{Large}}$ + ICP	93.9	55.1	93.7	84.4	97.8	92.1	86.2
$\text{RoBERTa}_{\text{Large}}$ + ST	**96.7**	**60.4**	**95.7**	**87.7**	**97.8**	**93.3**	**88.6**

Table 2: Results of self-training on natural language understanding benchmarks. We report a strong RoBERTa-Large baseline, as well as in-domain continued pretraining of this model (ICP) and our self-training approach (ST).

Model	SST-2	SST-5	CR	IMP	TREC	NER	Avg
Num samples	40	100	40	40	120	200	-
$\text{RoBERTa}_{\text{Large}}$	83.6±2.7	42.3±1.6	88.9±1.7	77.3±2.8	90.9±2.5	49.0±1.7	72.0±2.2
$\text{RoBERTa}_{\text{Large}}$ + ST	**86.7±2.3**	**44.4±1.0**	**89.7±2.0**	**81.9±1.4**	**92.1±2.4**	**58.4±1.4**	**75.5±1.8**

Table 3: Results of self-training for few-shot learning, using only 20 samples per class.

KL divergence. We found that fine-tuning again on the training set of the downstream task with ground-truth labels was not necessary, neither was adding ground-truth sentences from the training set to the self-training data.

Few-shot learning experiments. We sample 5 training sets that each consist of 20 examples from each label from the original training set of the task. We sample 200 examples from the original validation set of the task, taking the label distribution into account. We use the original test set of the task as our test set. For all experiments, we run 10 seeds for each train set and consider the mean test accuracy of top 3 models (based on their validation accuracy) as the performance on that train set. Based on this, we calculate the mean and standard deviation across 5 training sets, to report our final results. We synthetically annotate both retrieved and ground-truth data, and train each model for 50 epochs. Different from our experiments in the full-shot setting, we (1) use discrete labels, (2) include ground truth data in the training set, and (3) augment the reduced training set by one order of magnitude data samples sampled from the top 1000*(total supervised examples). These choices were made for few-shot learning experiments as the teacher model is not as strong, leading to noisier annotations compared to the full dataset setup.

4 Analysis and Results

In this section, we first report results on self-training, knowledge-distillation and few-shot learning with our best approach. We then provide an analysis of the key factors that makes self-training with SentAugment work in the context of natural language understanding.

4.1 Self-training experiments

In Table 2, we report results using self-training on six different downstream tasks. To understand the contribution of domain-adaptation and the actual contribution of self-training (ST), we compare ST to in-domain continued pretraining (ICP) where we continue masked language model pretraining of a RoBERTa-Large model on the retrieved in-domain augmented data. The goal of this comparison is to understand whether self-training only does domain adaptation to the target domain of the downstream task, which ICP also does. Indeed, RoBERTa-Large has been trained on a very large generic dataset of web data but not particularly specific to each downstream task.

First, we observe that self-training alone improves performance over a strong RoBERTa-Large baseline, leading to an 1.2% improvement on average. Improvements are largest on SST-5 and IMP, with 2.6% and 3.1% improvements respectively. On the other hand, when continuing pretraining on the self-training data with ICP, we observe a decrease in performance from 87.4% to 86.2%. It is interesting to note that this is not only the use of the in-domain data that is useful but the combination with the self-training algorithm. While ICP performs domain adaptation at pretraining time of the RoBERTa-Large model, it does not outperform the baseline. Self-training is thus a nontrivial way of improving generalization and doing domain-

Model	KD-data	SST-2	SST-5	CR	IMP	TREC	Avg
Models trained directly on the training set of each downstream task							
$\text{RoBERTa}_{\text{Large}}$	-	**96.5**	**57.8**	**94.8**	**84.6**	**97.8**	**86.3**
$\text{RoBERTa}_{\text{Small}}$	-	92.0	49.0	88.7	83.8	96.4	82.0
Models distilled using the same number of sentences as in the train set (cf. Table 1)							
$\text{RoBERTa}_{\text{Small(Large)}}$	GT	92.4	49.7	89.6	84.4	96.6	82.5
$\text{RoBERTa}_{\text{Small(Large)}}$	RD	90.7	47.5	87.4	69.1	90.8	77.1
$\text{RoBERTa}_{\text{Small(Large)}}$	SA	91.8	50.7	88.2	84.6	94.4	81.9
Models distilled using more unsupervised sentences (100k sentences)							
$\text{RoBERTa}_{\text{Small(Large)}}$	RD	92.5	51.2	92.4	78.1	96.2	82.1
$\text{RoBERTa}_{\text{Small(Large)}}$	SA	**94.2**	**57.6**	**92.6**	**85.5**	**97.0**	**85.4**

Table 4: Results of knowledge-distillation using ground-truth (GT), random (RD), or data-selected data (SA) as unnanotated sentences. We distill a RoBERTa-Large model of 24 layers into a RoBERTa-Small model with $100\times$ less parameters.

adaptation at fine-tuning time. (Xie et al., 2019) however show gains using ICP. We attribute that difference in our conclusion to (i) RoBERTa being trained on much more data than their BERT model trained on Wikipedia, (ii) our ICP using only approximately in-domain data rather than ground-truth.

4.2 Few-shot learning experiments

We investigate the effectiveness of our approach in the context of few-shot learning. In Table 3, we fine-tune a RoBERTa-Large model on between 40-200 samples of training data in each task and use it as a teacher model. Self-training leads to 3.5% average gains on all tasks, going from 72.0% to 75.5% while also reducing the variance. Gains are particularly strong on sequence labeling, where the student model obtains 58.4 F1 over 49.0 F1 for the teacher model.

4.3 Knowledge distillation experiments

Knowledge distillation (KD) also strongly benefits from large-scale augmentation. Table 4 shows baseline results from the RoBERTa-Large and RoBERTa-Small directly fine-tuned on the training set of each downstream task. Comparing distilled models that use different kinds of unannotated data, we observe that using the ground-truth (GT) leads to significantly better performance compared to random (RD) sentences, going from 77.1% to 82.5%. This shows that assuming the existence of data in the exact same domain is a strong assumption. Using the same amount of data, our data augmentation (SA) method bridges the gap with 81.9% average accuracy.

When leveraging more unannotated sentences, we push the random baseline to 82.1% which corresponds to a 5% improvement, getting closer to the GT baseline. Finally, using SentAugment leads to strong improvements, up to 85.4% average accuracy, only 0.9% average accuracy below the teacher model with almost ten times less parameters, showing the importance of data augmentation for KD.

4.4 Ablation study of data augmentation

Our approach leverages several key components that make data augmentation work and that enable self-training for natural language understanding. We examine these components in this section.

Task-specific retrieval. We compare different methods for building task-specific embeddings used as queries for retrieving in-domain sentences from the large bank of sentences. In Table 5, we observe that using one query for each label (label-average) leads to better performance than having a single query embedding for the entire task (all-average), leading to a 83.1% accuracy on average. For tasks with unbalanced classes, this avoids an over-representation of the majority class, and also provides more diversity in the retrieved sentences. Interestingly, having one query embedding per sentence in the training set does not improve performance, except for named entity recognition where

the per-sentence approach leads to the best performance.

Model	Selection	$\mathcal{C}$	SST-5	CR	NER	Avg
RoBERTa$_{\text{Large}}$ + ST	all-avg	$\mathcal{O}(Md^2)$	60.0	94.7	92.8	82.5
RoBERTa$_{\text{Large}}$ + ST	label-avg	$\mathcal{O}(KMd^2)$	**60.4**	**95.7**	93.1	**83.1**
RoBERTa$_{\text{Large}}$ + ST	per-sent	$\mathcal{O}(NMd^2)$	60.1	95.4	**93.3**	82.9

Table 5: Impact of data augmentation technique. $\mathcal{C}$ is the complexity, M the size of the bank of sentences, K the number of labels (or clusters), N the size of the downstream training set and d the embedding size.

Sentence embedding space. Our data augmentation method is based on structuring a large external bank of text with a sentence embedding space. The sentence embedding method plays an essential role as shown in Table 6. We compare three embedding methods, the average of fastText (Mikolov et al., 2018) word embeddings (average-word2vec), the uSIF-ParaNMT embeddings (Ethayarajh, 2018b) and our own sentence encoder. We observe that uSIF-ParaNMT and para-embeddings - two sentence embedding methods that obtain state-of-the-art results on semantic textual similarity benchmarks - lead to stronger performance than the average-word2vec approach. Para-embeddings leads to the best performance and improves performance over uSIF by 0.4% on average.

Model	Embedding	dim	SST-5	CR	NER	Avg
RoBERTa$_{\text{Large}}$ + ST	avg-w2v	300	59.4	95.2	92.9	82.5
RoBERTa$_{\text{Large}}$ + ST	uSIF	300	59.9	95.0	93.1	82.7
RoBERTa$_{\text{Large}}$ + ST	SASE	256	**60.4**	**95.7**	**93.1**	**83.1**

Table 6: Impact of sentence embedding method: average-word2vec, uSIF with ParaNMT and SASE.

Scaling bank size. To demonstrate the importance of large-scale retrieval, we evaluate our method using an increasing amount of data for our bank, from fifty million sentences to five billion sentences (one hundred billion words). We observe a significant increase in performance from 50m to 1B in Table 7, but the improvement seems to saturate when going from 1B to 5B. However, the 5B external bank may however provide additional gains for tasks that are in rare domains and that can leverage the additional 4B sentences, which correspond to 342M additional CommonCrawl documents. Another effect of increasing the corpus size may be reducing diversity in the retrieved sentences. We leave experimenting with diversity-inducing enhancements to the retrieval for future work.

Model	#lines	#words	SST-5	CR	NER	Avg
RoBERTa$_{\text{Large}}$ + ST	50m	1B	59.5	95.4	92.8	82.6
RoBERTa$_{\text{Large}}$ + ST	250m	5B	59.5	**95.7**	92.9	82.7
RoBERTa$_{\text{Large}}$ + ST	1B	20B	**60.4**	**95.7**	**93.1**	**83.1**
RoBERTa$_{\text{Large}}$ + ST	5B	100B	60.0	95.3	**93.1**	82.8

Table 7: Impact of sentence bank size (number of lines and words) on self-training results.

Continuous labels. In Table 8, we show that using class probabilities as synthetic labels leads to significantly better performance, outperforming discrete synthetic labels by 0.9% on average. We found very little gain when using self-training with discrete labels, contrary to previously published results in computer vision (Yalniz et al., 2019; Xie et al., 2020). A difference with previous work in computer vision is the number of classes of the supervised data. In that context, discrete labels provide even less information to the student model than continuous class probabilities.

Model	label type	SST-5	CR	NER	Avg
RoBERTa$_{\text{Large}}$ + ST	discrete	59.1	94.7	92.8	82.2
RoBERTa$_{\text{Large}}$ + ST	logits	**60.4**	**95.7**	**93.1**	**83.1**

Table 8: Impact of label type on self-training results.

Computational cost of self-training. SentAugment data prefiltering reduces the amount of data to be annotated by the teacher model and also filters based on the target domain. Filtering based solely on classifier confidence is significantly more expensive computationally, as annotating 10000 sentences with RoBERTa-Large takes approximately 3 seconds on a Volta-32GB GPU. This means that annotating 1B sentences takes 83 hours on a single GPU and much longer for models of larger size such as T5 (Raffel et al., 2019) or GPT-3 (Brown et al., 2020). On the other hand, using SentAugment based on a few task-specific query embedding (label-average) takes one minute for scoring 1B sentences. By only selecting the first few million top sentences, or less, to synthetically annotate, this greatly reduces computational cost and allows to scale to a larger bank of sentences, which in turn allows for more domains to be considered. Note that similarity search can be further sped up significantly by using fast nearest neighbor search such as product quantization with inverted files (Johnson et al., 2019).

BioNLP query: A single gene on chromosome 7 makes a protein called the cystic fibrosis transmembrane conductance regulator (CFTR). **Nearest neighbor**: Cystic Fibrosis A mutation in the gene cystic fibrosis transmembrane conductance regulator (CFTR) in chromosome 7.
Financial Query: Google has entered into an agreement to buy Nest Labs for $3.2 billion. **Nearest neighbor**: In January Google (NASDAQ:GOOG) reached an agreement to buy Nest Labs for $3.2 billion in cash.
Hate-speech Query: *Average sentence embeddings of the "hateful" class of IMP* **Nearest neighbor**: fuzzy you are such a d* f* piece of s* just s* your g* d* mouth. – All you n* and s* are fucking ret*
Movie review Query: *Average sentence embeddings of the "bad movie" class of SST-5* **Nearest neighbor**: This movie was terribly boring, but so forgettable as well that it didn't stand out for how awful it was..
Product review Query: *Average sentence embeddings of the "positive" class of CR* **Nearest neighbor**: The phone is very good looking with superb camera setup and very lightweight.
Question type Query: *Average sentence embeddings of the "location" class of TREC* **Nearest neighbor**: Lansing is the capital city of which state?

Table 9: Examples of nearest neighbors using a per-sentence or label-average query from different domains.

5 Analysis of similarity search

In this section, we present the results of our SentAugment sentence embedding (SASE) method on semantic textual similarity (STS) benchmarks and present examples of retrieved sentence based on large-scale similarity search.

5.1 Sentence embeddings (SASE)

In Table 10, we compare our sentence embedding method to previous approaches including BERT (Mean) (Devlin et al., 2018), InferSent (Conneau et al., 2017), GenSen (Subramanian et al., 2018), USE (Cer et al., 2018), Sentence-BERT (Reimers and Gurevych, 2019), uSIF (Ethayarajh, 2018a), Charagram (Wieting and Gimpel, 2017) and BGT (Wieting et al., 2019b). On average, our embeddings outperform previous approaches by 0.2% on STS 2012 to 2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), and by 0.9% on STS-Benchmark (Cer et al., 2017).

Model	Semantic Textual Similarity (STS)						
	2012	2013	2014	2015	2016	Avg	STS-B
BERT (Mean)	48.8	46.5	54.0	59.2	63.4	54.4	-
InferSent	61.1	51.4	68.1	70.9	70.7	64.4	70.6
GenSen	60.7	50.8	64.1	73.3	66.0	63.0	-
USE	61.4	59.0	70.6	74.3	73.9	67.8	-
Sentence-BERT	66.9	63.2	74.2	77.3	72.8	70.9	-
uSIF-	68.3	**66.1**	**78.4**	79.0	-	-	79.5
Word, trigram	67.8	62.7	77.4	80.3	78.1	73.3	79.9
BGT	68.9	62.2	75.9	79.4	**79.3**	73.1	-
SASE (ours)	**69.7**	62.9	77.3	**79.8**	78.1	**73.5**	**80.8**

Table 10: Results of our sentence encoder (SASE) on STS benchmarks from 2012 to 2016 and on the test sets of the STS-Benchmark dataset, compared to previously published results. We report Pearson's r $\times$ 100.

5.2 Examples of large-scale similarity search

SentAugment uses large-scale similarity search combined with an embedding space with billions of sentences to find in-domain sentences. In Table 9, we show examples of nearest neighbors extracted from CommonCrawl based on sentence-level or label-level queries and for different domains such as biomedical, financial or hate-speech data. We see that retrieving nearest neighbors can lead to good paraphrases which either preserve the meaning or augment it with additional information. We also observe reformulation of the same input sentence. As for label-level queries, we observe that retrieved sentences match very well the domain of the downstream task. We also release as part of our work nearest-neighbor indexes for researchers to explore further large-scale similarity search of web data. These indexes provide more examples of how well the model performs when trying to find similar sentences in our corpus using our sentence embedding. We hope this will lead to an improved understanding of large-scale embedding spaces and also help the community analyze the content and biases of large-scale web corpora used to train language models.

6 Conclusion

Recent work in natural language understanding has focused on unsupervised pretraining. In this paper, we show that self-training is another effective method to leverage unlabeled data. We introduce SentAugment, a new data augmentation method for NLP that retrieves relevant sentences from a large web data corpus. Self-training is complementary to unsupervised pre-training for a range of natural language tasks and their combination leads to further improvements on top of a strong RoBERTa baseline. We also explore knowledge distillation and extend previous work on few-shot learning by showing that open domain data with SentAugment is sufficient for good accuracy.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M Cer, Mona T Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *SemEval@ NAACL-HLT*, pages 252–263.

Eneko Agirre, Carmen Baneab, Claire Cardiec, Daniel Cerd, Mona Diabe, Aitor Gonzalez-Agirre, Weiwei Guof, Rada Mihalceab, German Rigaua, and Janyce Wiebeg. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. *SemEval 2014*, page 81.

Eneko Agirre, Carmen Baneab, Daniel Cerd, Mona Diabe, Aitor Gonzalez-Agirre, Rada Mihalceab, German Rigaua, Janyce Wiebef, and Basque Country Donostia. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. *Proceedings of SemEval*, pages 497–511.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-agirre, and Weiwei Guo. 2013. sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *In *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics*.

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of Semeval-2012*, pages 385–393.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations*.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Ondrej Bojar and Ales Tamchyna. 2011. Improving translation model by monolingual data. In *Workshop on Statistical Machine Translation (WMT)*.

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

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *NeurIPS*.

Mathias Johan Philip Creutz et al. 2018. Open subtitles paraphrase corpus for six languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087.

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

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the International Workshop on Paraphrasing*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proc. of EMNLP*.

Kawin Ethayarajh. 2018a. Unsupervised random walk sentence embeddings: A strong but simple baseline. *ACL 2018*, page 91.

Kawin Ethayarajh. 2018b. Unsupervised random walk sentence embeddings: A strong but simple baseline. In *Proceedings of The Third Workshop on Representation Learning for NLP*.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*, pages 168–177.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Jacob Kahn, Ann Lee, and Awni Hannun. 2020. Self-training for end-to-end speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7084–7088. IEEE.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.

Pierre Lison, Jörg Tiedemann, Milen Kouylekov, et al. 2019. Open subtitles 2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *LREC 2018, Eleventh International Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA).

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

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proc. of NAACL*.

Tomáš Mikolov, Édouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Daniel S Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V Le. 2020. Improved noisy student training for automatic speech recognition. *arXiv preprint arXiv:2005.09629*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *arXiv*.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

H Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

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 *EMNLP*, pages 1631–1642.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*.

Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Edouard Grave, Tatiana Likhomanenko, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. 2019. End-to-end asr: from supervised to semi-supervised learning with modern architectures. *arXiv preprint arXiv:1911.08460*.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207. ACM.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzman, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. *ICLR*.

John Wieting and Kevin Gimpel. 2017. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732*.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019a. Simple and effective paraphrastic similarity from parallel translations. *arXiv preprint arXiv:1909.13872*.

John Wieting, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019b. A bilingual generative transformer for semantic sentence embedding. *arXiv preprint arXiv:1911.03895*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *Proceedings of the 2nd Workshop on Evaluating Vector-Space Representations for NLP*.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698.

I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. 2019. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. 2020. Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*.