Semi-Supervised Learning with Unsupervised Data Augmentation in Text Classification

Tao Chen TC3404@NYU.EDU
Courant institute of Mathematical Sciences
New York University
New York, NY 10012, USA

Yuan Chen YC5588@NYU.EDU
Courant institute of Mathematical Sciences
New York University
New York, NY 10012, USA

Editor:

Abstract

The scarcity of labeled data remains a problem in the filed of Natural Language Processing since the last decade while research has shown great promises of consistency training on unlabeled data for the purpose of semi-supervised learning. Recent papers(Xie et al. (2019), Zhang et al. (2021), Sohn et al. (2020)) proposed advanced data augmentation methods for a good source of consistency training in text classification, and in this paper we focus on the performance and comparison between two augmentation methods: back-translation and word replacing. [1]

Keywords: consistency training, back-translation, word replacing

1. Introduction

Unlabeled data is common in Natural Language Processing. For examples, movie reviews without rating, so sometimes it is hard to figure out whether this review is positive or negative; news articles that are hard to classify into which category, as their content may be very broad. However, most of the deep learning architectures rely on labeled data, so this gives rise to semi-supervised learning, which is one of the most promising paradigms of leveraging unlabeled data to address the issue of scarcity of labeled data.

Research has shown great promises of consistency training on unlabeled data to regularize model predictions invariant to noise. The intuition behind this idea is that a good model should be robust to any small change in an input examples or hidden state. A few examples of noise injection include additive Gaussian noise, dropout noise, and adversarial noise.

Recently it has been pointed out that using advanced data augmentation methods to perform consistency training is a better source of noise in semi-supervised learning. There are two augmentation methods in the scope of text classification: back-translation and word replacing. Back-translation is to translate words in one language into another language and then translate back into the original language, while word replacing is to keep the key

1. Code is available at https://github.com/AbigailCY/semi-supervised-learning-text.git

words, but replace the uninformative words with other uninformative words, in this way to introduce the noise.

In the Unsupervised Data Augmentation paper, back-translation is shown to outperform purely supervised model which uses orders of magnitude more data, but the performance of word replacing is missing. In this report, we perform experiments on back-translation and word replacing, and track model performance with respect to the proportion of labeled and augmentation data; meanwhile investigate how TF-IDF score, a statistical measure that evaluates how relevant a word is to a document in a collection of documents, can affect model performance.

2. Background and Methods

2.1 Background: supervised data augmentation versus unsupervised data augmentation

Data augmentation is to create realistic-looking training data by applying a transformation to an example, without changing its label. Represent the augmentation transformation as $q(\hat{x}|x)$, from which we can draw augmented examples $\hat{x}$ based on an original example x. It is required that any sample $\hat{x}$ has to share the same label as x for an data augmentation to be valid.

Supervised data augmentation can be equivalently regarded as creating an augmented data set from the original supervised set and then training the model on the augmented set. It is crucial how to design this augmentation transformation as it is important for the augmented set to provide additional inductive biases. Supervised data augmentation provides a steady yet limited performance boost due to the fact that the labeled set is usually of a small size. Motivated by this limitation, and under the consistency training framework, unsupervised data augmentation plays an important role.

Figure 1 presents a recent workflow in semi-supervised learning with unsupervised data augmentation to enforce the smoothness of the model, where M in the figure represents a model that predicts a distribution of y given x. The general steps are: a. Given an input x, compute the output distribution $p_\theta(y|x)$ and a noised version $p_\theta(y|x, \epsilon)$ by injecting a small noise ϵ; b. Minimizing a divergence metric between the two distributions $p_\theta(y|x)$ and $p_\theta(y|x, \epsilon)$. This procedure enforces the model to be insensitive to noise and thus smoother with respect to changes in the input. How the noise is injected to the input can influence the performance of this consistency training framework. To enforce consistency, prior work generally employ simple noise injection such as adding Gaussian noise, but it has been shown that more advanced data augmentations that are more diverse and natural can lead to more significant performance gain under this setting.

2.2 Augmentation strategies: back-translation and word replacing

Back-translation refers to the procedure of translating an existing example x in language A into another language B and then translating it back into A to obtain an augmented example $\hat{x}$. It can generate diverse paraphrases while preserving the semantics of the original sentence, and the diversity of paraphrases is important in performance improvement. Figure 2 gives an example of back-translation.

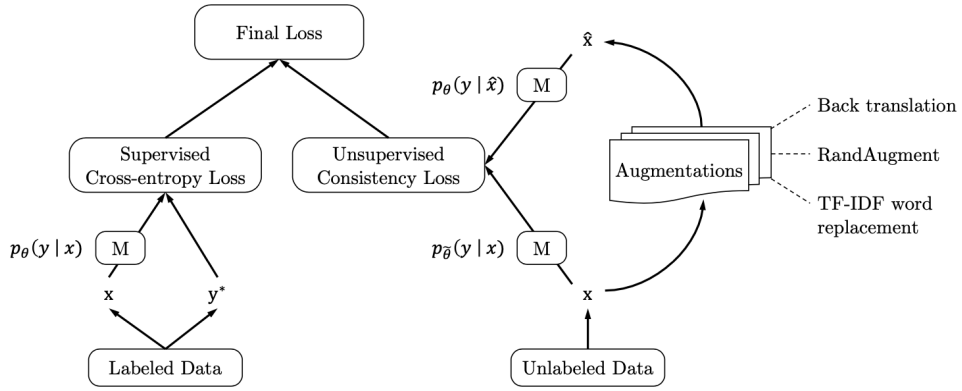

Figure 1: UDA diagram

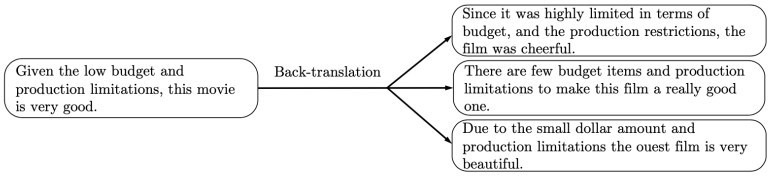

Figure 2: An example of back-translation

Back-translation is good at maintaining the global semantics of a sentence, but little control over which words will be retained, although this requirement is important for text classification. Therefore, word replacing is proposed which replaces uninformative words with low TF-IDF scores while keeping those with high TF-IDF scores. When a word is replaced, we sample another word from the whole vocabulary for the replacement, and the sampled words should not be keywords to prevent changing the ground-truth labels of the sentence.

3. Experiments

BERT will be used as our base model, which contains 12 layers, 12 attention heads, 768 hidden layers, and 110 million model parameters. Both the labeled data and unlabeled data come from one dataset, so this semi-supervised model is considered completely in-domain and serves as a good starting point as a baseline comparison.

There are two datasets in our experiments. Both are news articles from various topics. Dataset 1 (AGnews) covers 4 classes, with 30000 training samples in each class, and dataset 2 (20newsgroup) includes 20 classes, 600 training samples for each. The metrics used for model evaluation is topic classification accuracy. The detailed variants of experiment settings are shown below.

3.1 Preprocess

When looking into details of the original data, AGnews contains news title and a description, which is well organised, containing almost no junk data or characters, so no preprocessing is needed. UDA model will do the tokenization using bert base vocab file before training for both labeled, unlabeled, and test data. However, when coming into 20newsgroup dataset, it is composed newsgroup documents, which is long and poorly organised (containing junk information), so we used Gensim packages with stopwords to clean the text before doing tokenization. Experienment shows a difference before and after Gensim preprocessing – baseline (supervised learning with limited labeled data) without Gensim preprocessing can only achieve a classification accuracy of 0.582, but after Gensim preprocessing it can achieve a classification accuracy of 0.664. So we adopt this method in future comparison.

3.2 General training setting

To test the performance of semi-supervised learning, a baseline is created by supervised learning using only limited data. All the experiments are tested in Google cloud TPU and the training epoches are set to be at least 10. However, if the training steps is set too large, it would result in overfitting thus reduces the test accuracy. Also, up to a certain number of training steps, the KL divergence loss introduced by unlabeled and augmented data may be zero, meaning the training is useless from then. The training time ranges from 30 mins to several hours depending on the datasize. Also, max sequence length is set to be 128 tokens due to memory constraints, so the preformance is expected to be downgraded, especially on 20newsgroup dataset. The training learning rate envokes a linear decay. For UDA training, in each batch, the ratio between labeled and unlabeled data fed in to network is $1:3$.

3.3 Augmentation data size

For the 20newsgroup dataset, there are 11314 news paragraphs for training, so we adopt the total training set for augmentation. For AGnews dataset, the training set contains up to 120,000 samples, which is huge, so we tracked the model performance with increasing augmented data size. Figure 3 shows the UDA classification accuracy under different augmented size ranging from 10,000 to 60,000 with 1000 labeled data totally, under same number of epoches. As expected, it achieves a higher accuracy as the more augmented data fed in, so we adopt the 60,000 augmentation data in the following analysis for AGnews.

3.4 Augmentation method

We used three methods – unif, tf-idf and back translation in the experiments. There is a hyperparameter of the augmentation method ranging from 0 to 1, referring the magnitude of probability scores in word replacement. A higher value of hyperparameter means more words are replaced, or more noise introduced in to augmented data. The results for both datasets are shown in Table 1.

1. unif: unif is uniformly random selection on words to replace, compared to tf-idf. We tested the unif on 20newsgroup, and it turns out that the performance is worse than the other methods.

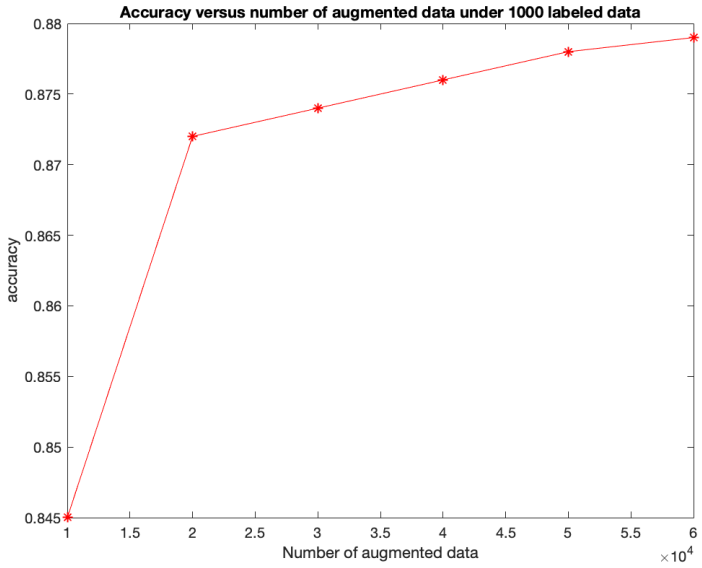

Figure 3: Accuracy versus number of augmented data, under 1000 labeled data, dataset 1: AGnews

2. tf-idf: The hyperparameter also influences the classification accuracy. With poor hyperparameter chosen, the model may not even outperform the baseline model, since the augmentation has no contribution to diversity. Experiment shows both datasets tends to adopt a hyperparameter of 0.9 and 0.7 respectively, where we need to balance between the diversity and validity of the augmentation data. Intuitively, with a higher hyperparameter, the data is more robust to noise. But in real experiments, empirical testes still need to be conducted to find the best augmentation method and hyperparameter.

3. back translation: Since the checkpoints used for back translation in UDA paper is not valid anymore, we used another approach to do back translation, namely the nlpaug library in python [nlp] . Even with gpu, this method is super slow, which takes almost 10 hours to translate 10k data, it is only applied to 20newsgroup. It turns out not better than tf-idf0.7. There are several reasons upon that, one is the max sequence length may limit the performance of bt, another reason is this back translation method may not introduce too much noise as a comparison.

Additionally, all these models doesn't outperform the fully-supervised case (using all training set as labeled data), which is understandable, because only a small bunch of labeled data are used. But in the UDA paper, by using another fine-tuned initalization base, namely BERT_FINETUNE instead of BERT_BASE, the accuracy of UDA can beat the fully-supervised case. This suggests further work on BERT_FINETUNE if having less computing resources limitation.

Dataset	augmentation method	Number of labeled data	baseline accuracy	SSL accuracy
AGnews	tf_idf-0.9	1000	**0.866**	**0.879**
AGnews	tf_idf-0.8	1000		0.874
AGnews	tf_idf-0.7	1000		0.856
AGnews	tf_idf-0.6	1000		0.844
AGnews	tf_idf-0.8	40	**0.744**	**0.851**
AGnews	tf_idf-0.9	40		0.844
AGnews	tf_idf-0.7	40		0.799
AGnews	No	all (purely supervised)	0.94	
20newsgroup	tf_idf-0.7	500	**0.657**	**0.710**
20newsgroup	tf_idf-0.6	500		0.705
20newsgroup	tf_idf-0.9	500		0.700
20newsgroup	unif-0.7	500		0.661
20newsgroup	back translate	500		0.67
20newsgroup	(fixmatch)tf_idf-0.9	500		0.699

Table 1: Comparison between baseline and UDA under different augmentation

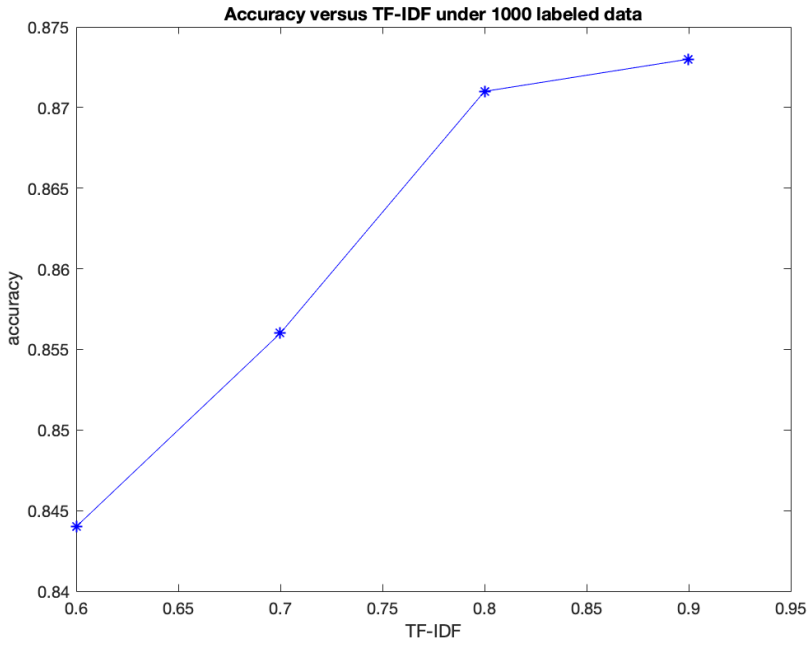

Figure 4: Accuracy versus TF-ITF, under 1000 labeled data, dataset 1

3.5 Empirical research on Fixmatch

Fixmatch Sohn et al. (2020) is another SSL model derived from UDA. Several minor modifications has been applied:

1. loss: It treats the original unlabeled data and augmented data as weak-augmented and strong-augmented data. And the unlabeled loss has been transformed from kl divergence for log probabilities (in UDA) to psuedo-labeling cross-entropy loss.

2. learning rate decay: Both linear rate decay and cosine learning rate decay are tested, and the cosine decay performs better in the paper.

3. optimizer: standard SGD with momentum gives better performance, whereas UDA uses Adam optimizer.

In Fixmatch paper, the author only proposed experiments and codes on image data. We want to study whether the psuedo-labeling cross-entropy loss is better than the one in UDA, so we tested it in the 20newsgroup data. Since we are using the bert base as pre-trained model, and it is trained with Adam optimizer, using SGD will not help to increase the accuracy. Instead, it means to train the whole network from the beginning. We tested using the SGD optimizer, and the training process has run for over 12 hours but the classification is still around 55%, so Adam is used in the bert base model here. We adopt the cosine learning rate decay as suggested in the paper.

With the same augmentation method tf_idf-0.9, shown in Table 1, Fixmatch achieves similar accuracy as UDA, so the psuedo-labeling loss also works in this case.

4. Conclusions

We performed the semi-supervised learning with advanced data augmentation on news classification, where the data augmentation includes back-translation and word replacing, and compare them to the baseline model using BERT. Generally speaking, the larger size of the augmented data and the higher TF-ITF value result in higher accuracy, with word replacing performing better than back-translation. For fixmatch, since the pseudo labeling requires a difference between 'strong' and 'weak' augmentation, we suggest to try out more augmentation methods and see which one applies to get a better performance in the future.

Acknowledgments

We would like to acknowledge support for this project from CSCI-GA 2565 Machine Learning class by Professor Rajesh Ranganath and multiple help from Mark Goldstein.

References

nlpaug library. URL https://github.com/makcedward/nlpaug.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *CoRR*, abs/2001.07685, 2020. URL https://arxiv.org/abs/2001.07685.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation. *CoRR*, abs/1904.12848, 2019. URL http://arxiv.org/abs/1904.12848.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *CoRR*, abs/2110.08263, 2021. URL https://arxiv.org/abs/2110.08263.