Analysis of Weak Supervision on Procedural Text

Ahmed Sohail Anwari, Anan Schütt
University of Saarland
66123 Saarbrücken
s8ahanwa, s8anscue@stud.uni-saarland.de

Abstract

ProPara is a dataset of procedural texts with state change and location labels introduced in [8]. This dataset is then used in the paper to solve the supervised learning task of predicting the labels given the sentence. Unused in previous works, the dataset also includes unlabelled training samples. In this paper, we applied semi-supervised algorithms on the dataset and compared the results with the purely supervised counterpart. We also incorporated BERT [4] to benefit from transfer learning.

1. Introduction

The generation of high quality labeled training data is crucial for a supervised learning model. In the domain of natural language processing (NLP), generating training data that is generalizable over a variety of task is very difficult due to the complexity of natural language and oftentimes, the complexity of the task itself. This creates a need for task-specific training datasets. Since most NLP tasks have human performance as a benchmark, the creation of new datasets requires human participants for data labeling.

Crowd-sourcing platforms such as Amazon Mechanical Turk and Prolific serve as great platforms for researchers to reach out to human participants. However, generating large amounts (millions of samples) of data is not only extremely expensive but is also prone to human errors. Hence, the need for semi-supervised methods is apparent here.

In our exploration, we consider the ProPara dataset [8]. The task for which this dataset was created was the detection of state changes of entities in procedural text. The procedural texts were in the form of paragraphs that described scientific processes. Amazon Mechanical Turk was used to annotate the paragraphs. These human annotations marked the creation, destruction and movement of entities at various time steps (in this case, sentences) in the procedural paragraph. To reduce the chances of errors, each paragraph was annotated by five annotators. ProPara has 488 labeled and 877 unlabeled paragraphs.

The high number of unlabeled data instances in ProPara makes it an ideal candidate to test weak supervision and its effectiveness when labeled data is scarce. In this report, we make use of a weak supervision technique known as ”learning by association” to improve the performance of the native model described in [8]. Furthermore, we also explore the use of BERT [4] in conjunction with ”learning by association”.

The rest of this report is structured as follows. In section 2, the research related to the topic is highlighted. Section 3 discusses some experiments that we conducted and the results of these experiments are discussed in section 4. Finally, we conclude in section 5.

2. Related Works

A brief overview of the dataset, models and approaches used in this work is presented in this section.

2.1. ProPara

The ProPara dataset was introduced in [8]. It contains procedural texts explaining simple processes, e.g. how rain forms, what happens during a volcanic eruption. The answers to these questions come in the form of multiple simple sentences, forming a paragraph. Each sentence could have different entities. For each entity in each sentence in the labelled portion of the dataset, labels are also provided. The state change label specifies the type of action in the sentence for a certain entity, and can have the value of `NONE`, `CREATE`, `DESTROY`, or `MOVE`. The location label provides the location of the entity before and after the sentence. Location labels could be blank if it is not applicable (e.g. before-location in a create sentence), or if the location does not appear explicitly in the sentence. This sets up for a supervised learning task.

The same paper also proposed two models to solve this problem. One of them is ProLocal, which reads each sentence independently as GloVe embeddings [9], passes them through BiLSTM [6] and bilinear attention [3] layers, and ends with fully connected layers to make predictions. The location prediction is reformulated as a classification task by outputting BIO-tags over the sentence instead of the words themselves. The predictions from all the sentences are then linked together to form a valid chain of events through simple rules.

2.2. Pseudo-label

Pseudo-labeling is a simple method to tackle semi-supervised learning tasks proposed in [7]. It refers to the process of producing labels for unlabeled samples by using a model trained on labeled data, then using it to in turn further train the model. In its most basic form, one can train the model using supervised data, then after a certain timestep, keep adding pseudo-labeled data to the supervised data, choosing the unlabeled samples on which the model is most confident. This process pushes decision boundaries closer to low-density regions, which, according to the cluster assumption [2], improves generalisation performance.

2.3. Learning by association

Learning by association is another algorithm for dealing with semi-supervised learning classification tasks proposed in [5]. In contrast to pseudo-labeling, learning by association does not attach any label to the unlabeled samples. Instead, it is based off the assumption that many unlabeled samples would have similar-looking labeled samples, and can thus be grouped closer to each other in an embedding space. This is done by defining a loss function on batches of labeled and unlabeled samples.

The loss function consists of three parts. The first part, $\mathcal{L}_{ce}$, is the cross entropy loss calculated over only the labeled samples. To formulate the other two parts, we must first consider embeddings and walks between samples.

An embedding is a mapping from the input to some space, called the embedding space. Preferably, similar input samples should map to points that are close together in the embedding space. This will then create clusters of input samples in the same class. Let us consider a batch of labeled samples A and unlabeled samples B. By defining a distance measure on the embedding space, for example the Euclidean distance or the dot product, we can define P^{ab}, the probability of walking from labeled to unlabeled samples as a matrix of dimension $|A| \times |A|$. This matrix is calculated by taking the softmax of the distance to each of the samples, then generate the next point according to that distribution. Similarly, P^{ba} the probability of walking from unlabeled to labeled samples can be defined. Walking from a labeled sample, to an unlabeled sample, and back to a labeled one forms a round trip. Since embeddings from the same class should be close together, the starting and ending class of a round trip should have the same class. We can then define round trip probabilities $P^{aba} = P^{ab} \cdot P^{ba}$.

To encourage these desired walks, the two losses are defined. First, Walker loss helps random walks to stay in the same class, and equalises the probability of walking between each pair of samples from the same class. This is formally written as:

$$\mathcal{L}_{walker} = H(T, P^{aba}) \tag{1}$$

where H is the cross entropy function, T, the target distribution, which is uniform between samples from the same class.

$$T_{ij} = \begin{cases} \frac{1}{\#class(A_i)} & \text{if class}(A_i) = \text{class}(A_j) \\ 0 & \text{else} \end{cases} \tag{2}$$

where $\#class(A_i)$ is the number of samples in the batch with the same class as A_i.

The other loss is the Visit loss, which encourages every unlabeled sample to be visited from the labeled samples. The purpose is to use all the unlabeled samples available, not just the ones that are conveniently similar to labeled samples. The loss is defined as:

$$\mathcal{L}_{visit} = H(V, \frac{\sum_{i=1} |A| P_i^{ab}}{|A|}) \tag{3}$$

where $V_i = \frac{1}{|B|}$ and P_i^{ab} is the i-th row of the matrix P^{ab}, which is the transition probability from the i-th labeled sample to each unlabeled samples.

These two losses only address the clustering of samples from the same class. The usual cross entropy loss is also added to make the correct predictions. The model can then be trained to minimise the (weighted) sum of these three losses.

For neural networks, we can use the second to last layer as our embedding layer, since this layer is behind the bulk of the network, and close embeddings in the same class directly results in higher accuracy.

2.4. BERT

BERT [4] is a large and complex model which has produced state-of-the-art results in several NLP tasks, such as question answering. At its core, BERT makes use of transformers [11] which are attention based models. These transformers are trained bi-bidirectionally which enables it to grasp the context of a word based on the surrounding words, as opposed to previous models which were trained in one direction only (left to right). These transformer units are then stacked together to form the full BERT model.

In the original paper, BERT model was pre-trained using Masked Language Model (MLM) and Next Sentence Prediction (NSP) tasks. It was then fine-tuned for 11 NLP tasks (as described in [4]) and produced promising results.It is now a general practice to use BERT pre-trained model for any NLP task (such as sequence classification) and to fine-tune the weights according to the training data and task at hand.

3. Experiments

We carried out experiments to test the effectiveness of learning by association on a natural language task, and it in combination with a pretrained model, namely BERT. To simplify the task, we modified it to just predict the type of change in a sentence for each participant, without predicting the location spans.

We used two models in the experiments, one without BERT and one with BERT. For the former, we used ProLocal as detailed in [8], but with a further fully connected layer before the final classification to represent the embedding required by learning by association. The latter is an implementation of BERT taken from [1] with two fully connected layers to represent the embedding and the classification output.

For each combination of model (with, without BERT) and training setting (supervised, semi-supervised, number of un-labeled samples), we ran five trainings and evaluations with five seeds. We present the average and standard deviation of the accuracy on the test set. We did not freeze any weight.

In all the experiments, we used embeddings of size 20. In supervised learning, we implemented early stopping to halt the training if validation accuracy doesn't improve within 2 epochs, but only if the accuracy is higher than 43%. This is set because training might take a few epochs before starting to not predict only one class, which covers 42.8% in the validation set. In learning by association, we used batches with 32 labeled samples and 32 unlabeled samples unless stated otherwise. We halved the weight of visit loss during training (i.e. visit loss weighted by 0.5, walker loss weighted by 1.0) , to reduce importance of visiting every single unlabeled sample, which seems unachievable because of the small batch size. We stop training if validation accuracy doesn't improve after 100 iterations, and set the same threshold as supervised learning.

4. Results

The results are shown in tables 1 and 2. First let us focus on results from ProLocal. Intuitively, more labeled samples lead to higher accuracy. As for the section with learning by association, unlabeled samples seem only to help in cases with fewer labeled samples, although the trend is not so clear because of the small number of trials and possible noise in the results.

We have also tried to change training settings. The results show no clear impact on the accuracy from modifying the weights or changing the batch size.

To further investigate the effects of learning by association, we calculated the embeddings of test samples and some unlabeled samples, then visualised them using the t-SNE algorithm [10]. The plots are shown in figure 3. One very noticeable difference between these two plots is that in the supervised model, `CREATE` samples form a very clear cluster, whereas in the semi-supervised model, the separation isn't as clear. A possible explanation is that in the semi-supervised model, the unlabeled samples which lie on the border between the clusters exist, which then tries to pull the two clusters together, resulting in these embeddings. Such a vague clustering also explains the lower accuracy in semi-supervised models.

As for the BERT models, the results show success in leveraging transfer learning capabilities of BERT well, even when there are few labeled samples available. The highest accuracy is achieved through BERT, using all labeled samples and no unlabeled samples. Learning by association shows slight improvement in some cases, but more often seem to be detrimental to the performance.

5. Conclusions

Machine question answering is a complex and diverse task. Due to the diversity of some text types (for example, procedural texts) and some question types (for example, state changes of entities in a passage), it is often difficult to obtain a gold standard labeled dataset for each individual task. A good weak supervision method can reduce the need of a large labeled dataset. Our results showed improvement in performance when labeled and unlabeled data is used in certain proportions. However, due to the scope of this work, no conclusive statements can be made. This leaves an exciting avenue for future research, as the need for generalizable weak supervision methods is greater than it was ever before.

References

[1] Pytorch transformers. https://github.com/huggingface/pytorch-transformers. Accessed: 2019-09-23.

[2] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *AISTATS 2005*, pages 57–64. Max-Planck-Gesellschaft, Jan. 2005.

[3] Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the cnn/daily mail reading comprehension task. *CoRR*, abs/1606.02858, 2016.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[5] Philip Häusser, Alexander Mordvintsev, and Daniel Cremers. Learning by association - A versatile semi-supervised training method for neural networks. *CoRR*, abs/1706.00909, 2017.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[7] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

ProLocal	0 unlabeled samples	1024 unlabeled samples	3498 unlabeled samples
256 labeled samples	42.30 (1.47)	44.60 (4.72)	44.09 (4.67)
512 labeled samples	44.60 (1.91)	47.86 (2.77)	49.33 (3.47)
2244 labeled samples	58.46 (2.05)	55.40 (1.62)	57.12 (1.96)
ProLocal		3498 unlabeled samples visit loss weight 0.25 walker loss weight 0.5	3498 unlabeled samples batch size 128
256 labeled samples		44.79 (2.95)	44.09 (2.53)
512 labeled samples		46.96 (4.46)	46.65 (3.49)
2244 labeled samples		53.16 (4.33)	57.25 (2.35)

Figure 1. Validation accuracies from the trials on modified ProLocal. The values are presented as mean (standard deviation).

BERT	0 unlabeled samples	1024 unlabeled samples	3498 unlabeled samples
256 labeled samples	52.84 (4.18)	56.55 (1.64)	54.06 (4.60)
512 labeled samples	60.70 (1.77)	55.78 (5.59)	58.27 (4.20)
2244 labeled samples	**67.35 (1.89)**	65.94 (1.22)	63.64 (3.47)

Figure 2. Validation accuracies from the trials on modified BERT. The values are presented as mean (standard deviation).

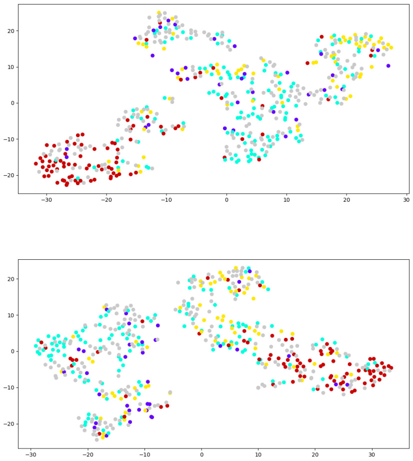

Figure 3. Top: embeddings of test samples (coloured by label: `NONE` in cyan, `CREATE` in red, `DESTROY` in yellow, and `MOVE` in blue) and unlabeled samples (grey) after t-SNE transformation, calculated by supervised ProLocal model. Bottom: same embeddings calculated by semi-supervised ProLocal model.

[8] Bhavana Dalvi Mishra, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. Tracking state changes in procedural text: A challenge dataset and models for process paragraph comprehension. *CoRR*, abs/1805.06975, 2018.

[9] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[11] 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.