The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

Semi-supervised Sentiment Classification with Dialog Data

Toru Shimizu[*1] Hayato Kobayashi[*1*2] Nobuyuki Shimizu[*1]

[*1]Yahoo Japan Corporation [*2]RIKEN AIP

The huge cost of creating labeled training data is a common problem for supervised learning tasks such as sentiment classification. Recent studies showed that pretraining with unlabeled data via a language model can improve the performance of classification models. In this paper, we take the concept a step further by using a conditional language model, instead of a language model. Specifically, we address a sentiment classification task for a tweet analysis service as a case study and propose a pretraining strategy with unlabeled dialog data (tweet-reply pairs) via an encoder-decoder model. Experimental results show that our strategy can improve the performance of sentiment classifiers and outperform several state-of-the-art strategies including language model pretraining.

1. Introduction

Sentiment classification is a task to predict a sentiment label, such as positive/negative, for a given text and has been applied to many domains such as movie/product reviews, customer surveys, news comments, and social media. A common problem of this task is the lack of labeled training data due to costly annotation work, especially for social media without explicit sentiment feedback such as review scores.

To overcome this problem, a framework [Dai and Le 2015] recently proposed a semi-supervised sequence learning framework, where a sentiment classifier based on recurrent neural networks (RNNs) is trained with labeled data after initializing it with the parameters of an RNN-based language model pretrained with a large amount of unlabeled data. The concept of their framework is simple but effective, and their work yielded many related studies of semi-supervised training based on sequence modeling, as described in Section 4.

In this paper, we take their concept a step further by using a conditional language model with unlabeled dialog data (i.e., tweet-reply pairs) instead of a language model with unpaired data[*1].

Contact: Toru Shimizu, Yahoo Japan Corporation, toshimiz@yahoo-corp.jp

∗1 We use the term "conditional language model" in a narrow sense only for a model trained with explicit source-target pairs, although both RNN-based language and autoencoder models can generate a text from a real-valued context vector.

An important observation of the dialog data that underpins our strategy is that the sentiment or mood in a message often affects messages in reply to it. People tend to write angry responses to angry messages, empathetic replies to sad remarks, or congratulatory phrases to good news.

Our contributions are listed as follows.

- We propose a pretraining strategy with unlabeled dialog data (tweet-reply pairs) via an encoder-decoder model for sentiment classifiers (Section 2.). To the best of our knowledge, our proposal is the first such proposal, as clarified in Section 4.
- We report on a case study based on a costly labeled sentiment dataset of 99.5K items and a large-scale unlabeled dialog dataset of 22.3M (Section 3.1).
- Experimental results of sentiment classification show that our method outperforms the current semi-supervised methods based on a language model, autoencoder, and distant supervision, as well as linear classifiers (Section 3.4).

2. Proposed Method

Our pretraining strategy simply consists of the following two steps:

1. Training a dialog (encoder-decoder) model using unlabeled dialog data (tweet-reply pairs)

as pretraining.

2. Training a sentiment classifier (encoder-labeler) model using labeled sentiment data (tweet-label pairs) after initializing its encoder part with the encoder parameters of the encoder-decoder model.

The encoder-decoder model is a conditional language model that predicts a correct output sequence from an input sequence [Sutskever et al. 2014]. This model consists of two RNNs: an encoder and decoder. The encoder extracts a context of the input sequence as a real-valued vector, and the decoder predicts the output sequence from the context.

Our classifier forms an encoder-labeler structure, which consists of the above encoder and a labeler that predicts a sentiment label from the context. Note that the encoder of the classifier is fine-tuned with labeled data, as in [Dai and Le 2015]. The main difference between their approach and ours is that we examine paired (dialog) data for pretraining, while they only showed the usefulness of pretraining with unpaired data.

3. Experiments

3.1 Datasets

We used two datasets, a dialog dataset for pretraining the encoder-decoder model and a sentiment dataset for training (fine-tuning) the sentiment classifier, as shown in Table 1.

The dialog dataset contains about 22.3 million tweet-reply pairs extracted from Twitter Firehose data.

The sentiment dataset includes about 100K tweets with manually annotated three-class sentiment labels: `positive`, `negative`, and `neutral`. Note that the tweets were sampled separately from those of the dialog dataset. Each tweet was judged by a majority vote of three experienced editors. The overall annotation work took roughly 300 person-days and was much more costly than collecting unlabeled dialog data. The breakdown of `positive`, `negative`, and `neutral` in the training set was 15.0, 18.6, and 66.4%, respectively. The average length of the tweets was 17 characters.

	Train	Valid	Test
Dialog	22,300,000	10,000	50,000
Sentiment	80,591	4,000	15,000

Table 1: Details of dialog and sentiment datasets

3.2 Model and Training

The settings of the dialog (encoder-decoder) model are as follows. In both the encoder and decoder, the size of the word-embedding layer is 256 and that of the LSTM-RNN hidden layer is 1024. The size of the output layer is 4000, which is the same as the (character-based) vocabulary size[*2]. The encoder and decoder share these hyper-parameters as well as the parameters themselves (that is, with regard to the embedding layer and recurrent layer). The total number of parameters is 8.9 million.

The settings of the sentiment classifier (encoder-labeler) model are as follows. The encoder part has the same structure and hyper-parameters as that of the dialog model, making them compatible for transferring learned parameters. We reused the dialog model's dictionaries in the classifier model so that the two models could process tweet texts consistently. The labeler consists of a fully connected layer and soft max nonlinearity.

The models were trained with ADADELTA with a mini-batch size of 64. The dialog model was trained in five epochs, and the classifier model was tuned with the early-stopping strategy, which stops training when the validation accuracy drops. For ADADELTA's parameters, we fixed the learning rate to 1.0, decay rate ρ to 0.95, and smoothing constant ϵ to 10^{-6} for all training sessions. We evaluated validation costs ten times per epoch and selected the model with the lowest validation cost. The training took 15.9 days on 1 GPU with 7 TFLOPS computational power.

3.3 Compared Models

We compared the following eight models: non-pretrained (`Default`), proposed dialog pretraining (`Dial`), current pretraining with unpaired data (`Lang`, `SeqAE`) and pseudo labeled data (`Emo2M`, `Emo6M`), and classical linear learners

*2 We used a character-based model since it performed better than word-based models in our preliminary experiments.

(LogReg, LinSVM). The details of these models are given below.

- Default: Trained without pretraining by executing only Step 2 in Section 2.
- Dial: Pretrained with the dialog model described in Section 2.
- Lang, SeqAE: Pretrained with the language model and autoencoder model proposed in [Dai and Le 2015]. The language model is the decoder part of the encoder-decoder model using a zero vector as the initial hidden layer value, and the autoencoder model is the same structure of the encoder-decoder model, where input and output are the same. Their training data were prepared with the same size as the dialog data for fare comparison.
- Emo2M, Emo6M: Pretrained with pseudo labeled data (2M, 6M) based on manually collected emoticons, which consist of 120 positive emoticons and 116 negative ones. This technique is also known as distant-supervision. These pseudo labels were annotated by extracting tweets including one of those emoticons, as in [Go et al. 2009]. The 2M dataset was created from 44.6M tweets in the training set of our dialog data. Since the 2M dataset is much smaller than the original dialog data, we prepared the 6M dataset additionally using another 92M tweets. Pretraining was conducted via a two-class sentiment classifier, which is a similar model to Default, since uncertain tweets without emoticons are not always neutral. We confirmed that this two-class classifier can reach more than 90% test accuracy on the emoticon-based test dataset. After pretraining, the parameters of the encoder part were transfered to the final classifier model.
- LogReg, LinSVM: Logistic regression and linear support vector machine (SVM) models of LIBLINEAR with bag-of-words features, which consist of 50K unigrams (w/o stopwords), 50K bigrams, and 233 emoticons. These features are based on a state-of-the-art system that performed best in the SEMEVAL competition. The best hyper-parameters were found through a grid-search on the validation set.

	5K	10K	20K	40K	80K
Default	68.47	71.48	72.86	75.07	76.50
Dial (ours)	**75.57**	**76.79**	**77.84**	**78.80**	**80.04**
Lang	74.49	75.51	76.80	78.04	79.26
SeqAE	70.53	72.34	73.45	75.18	76.46
Emo2M	67.71	68.88	70.47	73.08	75.75
Emo6M	67.79	68.47	70.42	72.72	74.86
LogReg	70.87	71.93	73.49	74.59	75.80
LinSVM	70.25	71.67	73.11	73.75	74.20

Table 2: Accuracy (%) of sentiment classification of each model versus labeled data size

3.4 Results

Table 2 shows the accuracy results of the compared models in Section 3.3 on the sentiment classification task when varying data size (5K to 80K). Each value is the average of five trials with different random seeds for each setting. The first row (Default) shows the default sentiment classifier model without pretraining. The second row block (Dial to Emo6M) shows the results of the same training as Default after pretraining via different models, while the third block shows those of linear classifiers (non-RNN models).

Comparing Dial with the other models, we can see that our pretraining strategy with dialog data consistently outperformed all the other models: state-of-the-art pretraining strategies with unpaired unlabeled data (Lang, SeqAE) as well as linear learners (LogReg, LinSVM). This indicates that unlabeled dialog data (tweet-reply pairs) have useful information for sentiment classifiers, as expected in Section 1. In fact, we confirmed that the pretrained encoder-decoder model can generate an appropriate reply, on which the sentiment on the input tweet is well reflected. For example, the reply ":(" was generated for the input tweet "I'm sorry to hear that".

Lang also outperformed well but did not overtake Dial. The differences between Dial and Lang are statistically significant*3 for all five training dataset sizes. Interestingly, SeqAE was not so effective like Dial, despite their model structures are basically the same. This implies that it is practically important to find appropriate data for pretraining, such as dialog data for sentiment classification.

*3 Under the significance level of 0.05 with two-tailed t-test assuming unequal variances.

As for the results of distant supervision with emoticons, both `Emo2M` and `Emo6M` performed worse than `Default`, and increasing the dataset size did not change the situation.

Comparing `Default` with `LogReg` and `LinSVM`, we can see that the linear models performed better than the default RNN model without pretraining, when the labeled data size is less than or equal to 20K. However, looking at the results of `Dial`, our method improved `Default` even for these cases (5K to 20K), and `Dial` clearly outperformed the linear models. This means that pretraining is useful especially on the situation where the labeled data size is limited.

4. Related Work

After [Dai and Le 2015] proposed the framework of semi-supervised sequence learning, there have been several attempts to extend sequence learning models for different tasks to semi-supervised settings. [Cheng et al. 2016] and [Ramachandran et al. 2017] studied semi-supervised training of machine translation models via an autoencoder model and language model, respectively. They also used paired data (parallel corpora), but unsupervised training was conducted with reasonable monolingual corpora to compensate for costly parallel corpora, which is opposite to our setting. [Zhou et al. 2016] proposed to use parallel corpora for adapting the sentiment resources in a resource-rich language to a resource-poor language. Their purpose was completely different from ours, since making parallel corpora is also costly.

5. Conclusion

We proposed a pretraining strategy with dialog data for sentiment classifiers. The experimental results showed that our strategy clearly outperformed the existing pretraining with unpaired unlabeled data via language modeling and pseudo labeled data via distant supervision, as well as linear classifiers. In the future, we will investigate whether or not we can use other paired data for pretraining of classification tasks.

References

[Cheng et al. 2016] Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-Supervised Learning for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*. Association for Computational Linguistics, pages 1965–1974.

[Dai and Le 2015] Andrew M Dai and Quoc V Le. 2015. Semi-supervised Sequence Learning. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, Curran Associates, Inc., pages 3079–3087.

[Go et al. 2009] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter Sentiment Classification using Distant Supervision. Technical report, Stanford Digital Library Technologies Project.

[Ramachandran et al. 2017] Prajit Ramachandran, Peter Liu, and Quoc Le. 2017. Unsupervised Pretraining for Sequence to Sequence Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. Association for Computational Linguistics, pages 383–391.

[Sutskever et al. 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, Curran Associates, Inc., pages 3104–3112.

[Zhou et al. 2016] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based LSTM Network for Cross-Lingual Sentiment Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*. Association for Computational Linguistics, Austin, Texas, pages 247–256.