Transfer-learning for Sentiment Classification of Hotel Reviews

Elozino Egonmwan
Department of Mathematics and Computer Science,
University of Lethbridge,
Lethbridge, AB, Canada.
elozino.egonmwan@uleth.ca

1 Introduction

Neural networks thrive on large amounts of parallel data, making it challenging for low-resource tasks. However, transfer-learning has proven to be especially useful in problems of this sort. In this experiment, the performance of two sentiment classification models fine-tuned from a supervised model versus an unsupervised classification algorithm are compared.

2 Methodology

2.1 Fine-tuning a transformer-based paraphrase classification model

The paraphrase classification task, labels two sentences as paraphrases (1) or not paraphrases (0) of each other. Good language and semantic understanding of the sentences is required for the model to accurately decide if one sentence contains similar information as the other (Zhang et al., 2020). Such semantic understanding is also fundamental to uncovering the sentiments underlying a given sentence, as presence of seemingly negative connotative words (e.g *terribly*) do not necessary make for a negative sentence (e.g the view from the balcony is *terribly fantastic*). Hence, the motivation to leverage knowledge learned from a paraphrase classification on sentiment classification of hotel reviews. The paraphrase classification model of choice is a BERT-based model (Devlin et al., 2018), trained on the General Language Understanding Evaluation (GLUE) task (Wang et al., 2018) and fine-tuned on Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005). For sentiment classification, the output (logits) of the model are fed to a fully-connected linear layer and a final softmax layer. This transfer-learning architecture, of training additional layer(s) on the output of a pretrained model is standard practice (Peng and Wang, 2020; Kim et al., 2015). Results of this model are presented in Tables 3 and 4.

2.1.1 Implementation Details

The model is set to continue training as long as there is an increase in the accuracy score on the validation set, in other words, early stopping (Li et al., 2019) is implemented when the accuracy on the validation set fails to improve after a set number of batches (50 in this case). The loss function minimized is the negative log likelihood loss. A batch size of 8 is used. This model was implemented using Pytorch on Google Colab .

2.2 Neural Laplace Naive Bayes Classifier

This forms a simple baseline model for experiment. The Naive bayes classifier is a supervised classification algorithm based on the Naive Bayes theorem, which assumes that features are statistically independent. It is a go-to algorithm for multi-classification problems especially where data is scarce. The naive bayes is used in this baseline as a feature estimator (Schneider, 2005). The features serve as input to a GRU-based linear classification model, topped by a softmax layer. This baseline model falls significantly short when compared to the fine-tuned model in section 2.1 because it is unable to learn sentence semantics. Rather it rests heavily on the bag of words assumption - where each word is represented independently of its context.

2.2.1 Implementation details

Tensorflow framework was employed in implementing this model. All sentences are preprocessed to remove punctuations and stop-words and are padded/truncated to a maximum of 140 words per sentence and 6 sentences per review. These decisions were based on the statistics of the dataset presented in Table 1.

2.3 Estimating Pseudo-labels for unlabelled held-out test set

In order to automatically estimate labels on the held-out test set to enable evaluation, algorithm 1 is used. Note that final evaluation scores for the test data with the final pseudo labels, $Pseudo_{final}$, are performed using MODEL$_M$ not MODEL$_{new}$

Algorithm 1 Generating Pseudo-labels for unlabelled test data- TEST$_d$

procedure PSEUDO LABELLING(MODEL$_M$, $Test_d$) ▷ Trained Model: MODEL$_M$; Unlabelled data: $Test_d$
  $Pseudo_1$ ← Predictions of MODEL$_M$ in inference mode on TEST$_d$
  $Train_{new}$ ← Concatenate the initial train set with $Pseudo_1$
  $Model_{new}$ ← Train MODEL$_M$ with $Train_{new}$
  $Pseudo_{final}$ ← Final predictions on $Test_d$ using $Model_{new}$
  return $Pseudo_{final}$
end procedure

2.4 Dataset statistics

The dataset contain $35,004; 7499; 6500$ valid samples in the training, validation and test sets respectively. Table 1 presents statistics of the dataset with represent to number of words and sentences per review. For the baseline model presented in section 2.2, we group reviews by ratings, and Table 2 shows the statistics.

Table 1: Statistics (Avg/Max) of the dataset samples after pre-processing.

	Train	Dev	Test
#sents/rev.	7/9	7/8	98/142
#words/sent	15/19	15/19	98/142
#words/rev.	98/142	15/19	98/142

2.5 Evaluation

The models are evaluated on Accuracy, Precision, Recall and F1 measure (see Tables 3 and 4 for results.

Table 2: Statistics of the dataset when grouped with respect to ratings

	# of reviews
rating=5	8453
rating=4	8542
rating=3	8341
rating=2	8311
rating=1	8257

Table 3: Accuracy, Precision, Recall and F1 evaluation scores on the test set

MODEL	ACC	PREC.	REC.	F1
Transformer				
NaiveBayes	81.21	82.82	81.21	82.00

Table 4: Accuracy, Precision, Recall and F1 evaluation scores on the dev set

MODEL	ACC	PREC.	REC.	F1
Transformer	76.45	77.19	76.45	76.81
NaiveBayes	35.09	35.13	35.09	35.11

References

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

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Young-Bum Kim, Karl Stratos, Ruhi Sarikaya, and Minwoo Jeong. 2015. New transfer learning techniques for disparate label sets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 473–482.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. 2019. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. *arXiv preprint arXiv:1903.11680*.

Peng Peng and Jiugen Wang. 2020. How to fine-tune deep neural networks in few-shot learning? *arXiv e-prints*, pages arXiv–2012.

Karl-Michael Schneider. 2005. Techniques for improving the performance of naive bayes for text classification. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 682–693. Springer.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue:

A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.