HW1: Classification

Noriyuki Kojima
nk654@cornell.edu

February 2020

1 Introduction

Text classification is a fundamental task for natural language processing (NLP), and its simplicity opens the door to feasible explorations to modern NLP technologies without massive computes. In this work, we build five different text classification systems and benchmark them on the popular Stanford Sentiment Treebank dataset (Socher et al., 2013). Our five systems are varied by underlying textual representations (i.e., unigrams, pre-trained word vectors (Joulin et al., 2016) and pre-trained contextualized features (Devlin et al., 2018)) and classification algorithms (i.e., naive-bayes, logistic-regression, feed-forward neural networks and convolutional neural networks (Kim, 2014)), and our goal is to implement and evaluate each variation. Our results suggest that the simplistic system remains competitive on this task, while most recent pre-trained contextualized features outperform everything.

2 Problem Description

In this section, we formally define the text classification task in Stanford Sentiment Treebank dataset (Socher et al., 2013). Given indices of common vocabulary $\mathcal{V}$ in English, let $s_i : w_{i,1}, ..., w_{i,N}$ be the i th sentence in the dataset, and $w_{i,j}$ is a index corresponding to the vocabulary of j th token in s_i. Our task is to build a generative system to model a categorical distribution of the binary sentiment label $y_i \in \{negative, positive\}$ for the sentence s_i, and the distribution has to be conditioned on $w_{i,1}, ..., w_{i,N}$ tokens in s_i. Formally, this is defined in Eq. 1.

$$y_i \sim Cat(f([w_{i,1}, ..., w_{i,N}])) \quad \text{where} \quad f : \mathbb{R}^N \rightarrow \Delta^1 \tag{1}$$

However, in many of modern practices in NLP, tokens are represented as vectors instead of indices. Therefore, we consider a embedding function E: $\mathbb{R}^N \rightarrow \mathbb{R}^{NxD}$, where j th row in the output embeddings of E is a D-dimensional vector representation for j th token in the input sentence s_i. We reformulate Eq. 1 to Eq. 2.

$$y_i \sim Cat(f(E([w_{i,1}, ..., w_{i,N}]))) \quad \text{where} \quad E : \mathbb{R}^N \rightarrow \mathbb{R}^{NxD}, f : \mathbb{R}^{NxD} \rightarrow \Delta^1 \tag{2}$$

In our work, we only consider three types of embedding function E.

- Unigram: $E([w_{i,1}, ..., w_{i,N}]) = [e_{i,1}; \ldots e_{i,1}; \ldots e_{i,N}]$, where $e_{i,j} \in \mathbb{R}^{|V|}$ is an one hot vector for the word $w_{i,j}$ for all $j \in \{1, \ldots N\}$.
- Word2vec: $E([w_{i,1}, \ldots, w_{i,N}]) = [\text{LUT}(w_{i,1}); \ldots; \text{LUT}(w_{i,N})] = [e_{i,1}; ...e_{i,1}; ...e_{i,N}]$, where LUT represents a look-up table which maps word-indices to corresponding pre-trained word embeddings $e_{i,j} \in \mathbb{R}^{300}$ for all $j \in \{1, \ldots N\}$ (Joulin et al., 2016).

- BERT: $E([w_{i,1}, ..., w_{i,N}]) = \text{BERT}([w_{i,1}, ..., w_{i,N}]) = [e_{i,1}; \dots e_{i,1}; \dots e_{i,N}]$, where BERT represents bi-directional 12-layer transformers (Vaswani et al., 2017) pre-trained for self-supervised objectives on massive un-annotated corpora. We take features $e_{i,j} \in \mathbb{R}^{768}$ for all $j \in \{1, \dots N\}$ from the final layer of transformers.

3 Model and Algorithms

To address the text classification task defined in the previous section, we build five classification systems as described below. All systems can be defined as $y_i \sim \text{Cat}(f(\text{E}([w_{i,1}, ..., w_{i,N}]), \Theta))$, and we denote the type of embedding function E inside a bracket of each system's name and describe f in a text description.

- Naive-Bayes(Unigram): f is a binarized naive bayes proposed by Wang and Manning (2012).
- Logistic Regression(Unigram): f is a logistic regression. Formally, we can denote $f([e_{i,1}; \dots e_{i,1}; \dots e_{i,N}]) = \sigma\left(W^T(e_{i,1} + \dots + e_{i,1} + \dots e_{i,N}) + b\right)$.
- Feedforward(W2V): This system is similar to CBoW model proposed by Mikolov et al. (2013). In our case, LUT of word2vec functions as a weight between an input and a projection layer. We sum resulting $[e_{i,1}; \dots e_{i,1}; \dots e_{i,N}]$ across tokens, and feed it to two-layered feed-forward networks followed by softmax.
- CNN(W2V): f is CNN architecture proposed by Kim (2014). The core idea is applying 2D convolutions of different kernel sizes on $[e_{i,1}; \dots e_{i,1}; \dots e_{i,N}]$ to extract features representing different temporal relations across tokens.
- Feedforward(BERT): f is two-layered feed-forward networks followed by softmax.

4 Experiments

4.1 Training

For all gradient-based methods (other than Naive-Bayes(Unigram)), we apply cross-entropy between target labels and predicted categorical distributions as our training objective. We use Adam optimizer with a training rate α_{lr}. For regularization, we employ dropout with a keep probability 0.5. For CNN(W2V), we further apply l2norm regularization with l_2 constant of 3 and λ of 0.01 (Kim, 2014). We train each system with k epoch, and pick a checkpoint with the best validation F1-score. We do grid-search on hyper-parameters and pick the hyper-parameters with the best validation F1-score. For Naive-Bayes(Unigram), we used the smoothing factor α_{sm} of 1. We summarize the optimal hyper-parameters for each system in Tab. 1.

Model	α_{lr}	k
NAIVE-BAYES(UNIGRAM)	—	1
LOGISTIC REGRESSION(UNIGRAM)	0.002	30
FEEDFORWARD(W2V)	0.0002	50
CNN(W2V)	0.0002	30
FEEDFORWARD(BERT)	0.00001	30

Table 1: Learning rates and training epochs used for optimal models in each system.

4.2 Results

We report the accuracy and F-score evaluated on the test set using our best models for each system. Our results are summarized in Tab. 2. From our result, Naive-Bayes(Unigram) performs surprisingly competitive while relying on primitive unigram textual embeddings and being only the non-gradient-based method. In contrast, systems using pre-trained word2vec perform a slight poorly in comparison to other systems. In addition, CNN classifier did not provide much gain in our experiment despite its complexity. Finally, the system with pre-trained contextualized representation BERT outperformes other systems by a large margin, aligning to the previous report (Devlin et al., 2018).

Model	Acc.	F-score
NAIVE-BAYES(UNIGRAM)	82.1	81.9
LOGISTIC REGRESSION(UNIGRAM)	81.6	81.4
FEEDFORWARD(W2V)	80.1	79.5
CNN(W2V)	81.8	81.3
FEEDFORWARD(BERT)	**90.1**	**90.2**

Table 2: Accuracy and F-score of each system in Stanford Sentiment Treebank Test Set

5 Conclusion

In this work, we build five text classification systems, varied by embedding function and classifiers, and we bench-marked the five systems on Stanford Sentiment Treebank data-set. Our finding suggests that the classical system, such as Naive-Bayes(Unigram), performs competitive, and we confirmed that the pre-trained contextualized representation is the most effective. However, pre-trained word2vec and complex CNN-based classifier did not provide much gain in comparison to more pre-mature systems. One possible cause is the lack of computes in our experiments. To accurately compare systems side-by-side, we need massive computes to cover the entire hyper-parameter space. This curse of computes seems to be more salient when it comes to training more complex modern deep learning systems.

References

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wang, S. and Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics.