Sentiment Analysis

IUST NLP {Custom} Project

Hasan Hammad
School of Computer Engineering
Iran University of Scince and Technology
h_hammad@comp.iust.ac.ir
401722334
NLP Final Project - Sentiment Analysis (link)

Abstract

Achieving accurate sentiment analysis is a crucial task in natural language processing, with applications ranging from product reviews to social media sentiment tracking. In this project, we delve into various embedding techniques coupled with both traditional classifiers and deep learning models to enhance sentiment analysis performance on the IMDB dataset. Our experimentation includes the exploration of diverse embedding methods. We investigate four basic classifiers, along with employing several text preprocessing techniques. Moreover, an attempt is made to address this issue by integrating BERT Embedding with two distinct deep neural network architectures. Our findings reveal that sentence embeddings play a pivotal role in significantly improving prediction accuracy, leading to notable increases in F1 score for the traditional classifiers. Furthermore, the integration of deep learning models proves beneficial, resulting in enhanced model performance. On the IMDB dataset, our best-performing basic classifier, SVM with BERT sentence embeddings, achieved an accuracy of 90.38. Furthermore, our most effective deep-learning model attained an accuracy score of 90.70.

1 Introduction

Sentiment analysis, a pivotal facet of natural language processing, plays a crucial role in deciphering the subjective tone and attitudes expressed in textual data. With applications spanning diverse domains such as product reviews and social media sentiment tracking, the accurate classification of sentiments has become increasingly indispensable.

Previous work uses various neural models to learn text representation, including convolution models (Kalchbrenner et al., 2014 in [1]; Zhang et al., 2015 in [2]; Conneau et al., 2016 in [3]; Johnson and Zhang, 2017 in [4]; Zhang et al., 2017 in [5]; Shen et al., 2018 in [6]), recurrent models (Liu et al., 2016 in [7]; Yogatama et al., 2017 in [8]; Seo et al., 2017 in [9]), and attention mechanisms (Yang et al., 2016 in [10]; Lin et al., 2017 in [11]).

Recently (Sun et al., 2020 in [12]) conducted extensive experiments to investigate the different approaches to fine-tuning BERT [13] for the text classification task. In this project, we embark on a comprehensive exploration of several techniques to enhance sentiment analysis performance, focusing specifically on the IMDB dataset [14].

The IMDB dataset, renowned for its diverse and extensive collection of movie reviews, serves as an ideal testbed for evaluating the efficacy of various embedding techniques and model architectures in sentiment classification. By delving into both traditional classifiers and cutting-edge deep learning models, our objective is to assess the performance of different methods in sentiment analysis.

Our project involves a meticulous investigation of diverse embedding methods, including the integration of BERT embeddings with distinct deep neural network architectures. The study extends beyond

embeddings, encompassing the scrutiny of four basic classifiers and the application of various text preprocessing techniques to address potential challenges in sentiment analysis.

As sentiments are nuanced and context-dependent, understanding the impact of sentence embeddings and the interplay between traditional classifiers and deep learning models becomes essential. Our findings aim to shed light on the pivotal role of embedding techniques in enhancing prediction accuracy, while also showcasing the synergies achieved through the integration of deep learning models.

Through rigorous experimentation and evaluation on the IMDB dataset, we aim to identify the most effective combination of embedding techniques and model architectures for sentiment analysis. Furthermore, this project contributes valuable insights into the optimization of sentiment analysis models, shedding light on the effectiveness of different embedding techniques and model architectures for accurate sentiment classification in the context of the IMDB dataset.

2 Related Work

The foundation of this project, as introduced by (Sun et al., 2020 in [12]), investigates the optimization intricacies of BERT for the specific task of text classification. BERT, a transformer-based model pretrained on diverse language tasks, has demonstrated exceptional success across various natural language processing applications.

However, fine-tuning BERT for text classification entails addressing nuanced challenges inherent to this domain. Acknowledging the significance of pre-trained word embeddings in modern NLP systems, exemplified by (Mikolov et al., 2013 in [15]) and (Pennington et al., 2014 in [16]), the project explores their potential improvements over scratch-learned embeddings.

Moreover, the generalization of word embeddings, including sentence embeddings (Kiros et al., 2015 in [17]; Logeswaran and Lee, 2018 in [18]) and paragraph embeddings (Le and Mikolov, 2014 in [19]), is examined as features in downstream models. (Peters et al., 2018 in [20]) notably enhance the state-of-the-art in major NLP benchmarks by concatenating embeddings derived from language models as additional features.

Beyond unsupervised pre-training, transfer learning with extensive supervised data has demonstrated notable performance in various tasks such as natural language inference (Conneau et al., 2017 in [21]) and machine translation (McCann et al., 2017 in [22]).

This project aims to assess multiple embedding methods with diverse classifiers, comparing the outcomes with two deep learning models exclusively utilizing BERT embeddings.

3 Approach

The primary approach in this study revolves around the incorporation of BERT Embedding into two distinct deep neural network architectures. This strategy is designed to harness the inherent capabilities of BERT Embedding in capturing nuanced contextualized word representations and amplifying them.

Additionally, the methodology proposed in this project encompasses the utilization of several simple classifiers alongside four distinct embedding methods. The intention is to systematically elucidate the strengths and weaknesses inherent in different embedding techniques.

3.1 Traditional Classifiers

In our project, we employed four fundamental base classifiers (SVM [23], Naive Bayes [24], Decision Tree [25] and Random Forest [26]), coupled with the implementation of various text preprocessing techniques, which will be elaborated upon later. The selected classifiers are widely recognized in the field of machine learning, and we applied them to categorize feature vectors derived from different embedding techniques. In this phase of the project, four embedding methods were utilized: BOW [27], TF-IDF [28], Word2Vec [29], and BERT. Furthermore, BERT was employed to generate two distinct embeddings—word embeddings and sentence embeddings.

3.2 Deep Models

We conducted experiments with two deep architectures utilizing BERT as the foundational embedding. The initial model involved the fusion of BERT with a Bi-Directional LSTM [30], while the second model entailed combining BERT with a Feed Forward Neural Network (FFN).

3.2.1 BERT with Bi-LSTM

This model incorporates a base model (BERT) for embeddings and extends it with a bidirectional LSTM layer. The model consists of an input layer, followed by a bidirectional LSTM layer with hidden size of 320. The output of the LSTM is then fed through a fully connected neural network (FC) comprising dropout regularization, three linear layers with decreasing sizes (320*2 to 80 to 20), and a final linear layer to classify the output.

3.2.2 BERT with FNN

Here we combined a the base model (BERT) with a Feed Forward Neural Network (FNN) for the binary classification task of sentiment analysis. The model consists of an input layer and an output layer. The base model's last hidden state is extracted for the [CLS] token, representing the entire input sequence. This representation is then passed through a dropout layer for regularization and a linear layer, followed by a sigmoid activation function to obtain the binary classification probabilities.

4 Experiments

4.1 Data

The IMDb dataset is a binary sentiment analysis dataset consisting of 50,000 reviews from the Internet Movie Database (IMDb) labeled as positive or negative. The dataset contains an even number of positive and negative reviews. Only highly polarizing reviews are considered. A negative review has a score ≤ 4 out of 10, and a positive review has a score ≥ 7 out of 10. No more than 30 reviews are included per movie. This dataset was compiled by Andrew Maas.

4.2 Data Pre-processing

As the dataset originates from web scraping, it contains HTML codes that need to be addressed. Cleaning the text involves the removal of HTML tags, elimination of numbers, punctuation, and stop words, substitution of negative contractions with their complete forms (e.g., "won't"), breaking down compound nouns formed with hyphens (except for BERT), and standardizing texts by converting them to lowercase.

For stop word removal, the NLTK stop words set was employed, with modifications. Words with negative connotations, such as "not" or "nor," were removed from the set, and contraction patterns like 're or 've were added. This customized stop word set was specifically applied for Word2Vec vectorization.

Considering that BERT embedding is trained on Wikipedia data, certain exceptions are made. Numbers and select punctuation marks [, / () : ; '] are retained in the text, along with compound nouns formed with hyphens, to ensure a more reliable embedding. Additionally, !, ?, and . are preserved to identify sentence endings for subsequent purposes, such as generating BERT embeddings on a per-sentence basis.

For BOW and TF-IDF embeddings, stemming and lemmatization are performed based on the part-of speech (POS) tags of words.

Finally, white spaces are replaced with a single space to streamline the text.

4.3 Evaluation method

When assessing the performance of the basic classifiers, we employed well-established metrics, namely accuracy and F1 score. These metrics, widely employed in contemporary research, offer a thorough evaluation of model performance by incorporating both precision and recall aspects in

our sentiment analysis task. The adoption of accuracy and F1 score aligns our evaluation approach with prevalent standards in the field, bolstering the reliability and comparability of our findings. Conversely, for the evaluation of the deep models, we exclusively relied on accuracy scores.

4.4 Experimental details

All our models were executed on Google Colab. For the basic classifiers, we utilized identical feature vectors derived from various embedding techniques, including BOW, TFIDF with stop words, TFIDF without stop words, Word2Vec with stop words, Word2Vec without stop words, BERT word embeddings, and BERT sentence embeddings. This led to the execution of seven experiments for each of the four basic classifiers, resulting in a total of 28 experiments.

Concerning the deep models, consistent hyperparameters were employed, namely a learning rate of 1e-5, weight decay set to 0.01, a training batch size of 4, and a test batch size of 6. Due to limited GPU resources, the training process was limited to 10 epochs for each deep model.

4.5 Results

As previously mentioned, our exploration encompasses various embedding methods. We scrutinize four fundamental classifiers while incorporating multiple text preprocessing techniques. Furthermore, we endeavor to address this challenge by integrating BERT Embedding with two separate deep neural network architectures.

4.5.1 Traditional Classifiers Results

In the phase dedicated to basic classifiers, given the substantial number of experiments conducted to assess their performance, we will present the top and second-top accuracy and F1 score achieved by each classifier, along with the associated embedding methods.

Table 1 provides a summary of the outcomes achieved by the basic classifiers.

Table 1: Traditional Classifiers Evaluation Results

Classifier	Embedding Method	Accuracy	F1 Score
Decision Tree	BOW	72.51	0.73
Decision Tree	BERT (Sentence)	78.47	0.78
Naive Bayes	BERT (Sentence)	84.22	0.85
Naive Bayes	TFIDF without stop words	84.58	0.85
Random Forest	TFIDF without stop words	85.10	0.85
Random Forest	BERT (Sentence)	86.22	0.86
SVM	TFIDF without stop words	87.93	0.88
SVM	BERT (Sentence)	90.38	0.90

The results obtained from the SVM classifier are not only promising but also exceed our expectations, primarily due to the incorporation of BERT sentence embeddings. Moreover, the findings underscore that certain straightforward methods can yield commendable outcomes, given the appropriate utilization of embeddings, despite their inherent simplicity.

4.5.2 Deep Models Results

During this phase, we ensured a fair comparison by training the deep models with identical hyperparameters. To account for limited GPU resources, the training process was restricted to a maximum of 10 epochs.

Table 2 details the highest test and train accuracy scores achieved by each model.

The unexpected results can be attributed to the limitation in training the models for an insufficient number of epochs. The simpler model (BERT + FNN) outperformed due to its fewer parameters, while the more complex model (BERT + LSTM) was unable to surpass these scores given the restricted training epochs.

Table 2: Deep Learning Models Evaluation Results

Method	Base	Train Accuracy	Test Accuracy
Bi-LSTM	BERT	96.33	89.10
FNN	BERT	96.60	90.70
withIn-Task Pre-Training [12]	BERT-Large	-	95.79

Acknowledging that training extensive models for a limited number of epochs is not ideal, the constraints imposed by limited GPU resources necessitated such compromises. Despite the non-ideal conditions, this experiment serves as a valuable evaluation of deep methods under computational constraints, providing a realistic assessment of their performance within the given limitations.

5 Analysis

5.1 Traditional Classifiers

SVM Classifier with BERT Sentence Embeddings:
Promising Performance: The SVM classifier demonstrated promising results, surpassing initial expectations. The utilization of BERT sentence embeddings appears to have significantly contributed to the enhanced performance.

Overall Simplicity and Effectiveness:
Surprising Efficacy: The results underscore the effectiveness of some simple classification methods, considering their simplicity. The key factor in achieving good results lies in the judicious choice of appropriate embeddings.

5.2 Deep Models

BERT + FNN Model:
Optimal Performance Given Constraints: Despite the constraint of limited training epochs due to GPU limitations, the BERT + FNN model showcased better results. Its fewer parameters and relative simplicity allowed it to outperform other models within the given computational constraints.

BERT + LSTM Model:
Challenges Due to Limited Epochs: The BERT + LSTM model faced challenges in surpassing scores, mainly due to the model's larger size and the restricted number of training epochs. The inherent complexity of the LSTM architecture requires more extensive training for optimal performance.

Evaluation Under Computational Constraints:
Realistic Assessment: Recognizing the non-ideal conditions imposed by GPU limitations, this experiment serves as a realistic evaluation of deep methods under such computational constraints. Despite these challenges, the outcomes provide valuable insights into the models' performance within practical limitations.

6 Conclusion

In conclusion, both stages of the project provide valuable insights. The Simple Classifiers emphasize the efficacy of fundamental classifiers when paired with thoughtfully selected embeddings. On the other hand, the Deep Models emphasize the importance of prudent model selection and training strategies, particularly within the constraints of computational resources. These revelations contribute to a nuanced comprehension of sentiment analysis models, considering both simplicity and complexity in their development and evaluation.

7 Links

The full implementation of this project can be found in the following link:

NLP Final Project - Sentiment Analysis (link)

References

[1] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

[2] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In Advances in neural information processing systems, pages 649–657.

[3] Alexis Conneau, Holger Schwenk, Loıc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for natural language processing. arXiv preprint arXiv:1606.01781,2.

[4] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, pages 160–167. ACM.

[5] Yizhe Zhang, Dinghan Shen, Guoyin Wang, Zhe Gan, Ricardo Henao, and Lawrence Carin. 2017. Deconvolutional paragraph representation learning. In Advances in Neural Information Processing Systems, pages 4169–4179.

[6] Dinghan Shen, Yizhe Zhang, Ricardo Henao, Qinliang Su, and Lawrence Carin. 2018. Deconvolutional latent-variable model for text sequence matching. In Thirty-Second AAAI Conference on Artificial Intelligence.

[7] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101.

[8] Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom. 2017. Generative and discriminative text classification with recurrent neural networks. arXiv preprint arXiv:1703.01898.

[9] Minjoon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Neural speed reading via skimrnn. arXiv preprint arXiv:1711.02085.

[10] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1480–1489.

[11] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130.

[12] Sun, C., Qiu, X., Xu, Y., Huang, X. (2019). How to Fine-Tune BERT for Text Classification?. In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds) Chinese Computational Linguistics. CCL 2019. Lecture Notes in Computer Science(), vol 11856. Springer, Cham. https://doi.org/10.1007/978-3-030-32381-3_16

[13] Devlin, Jacob, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." North American Chapter of the Association for Computational Linguistics (2019).

[14] Maas, Andrew & Daly, Raymond & Pham, Peter & Huang, Dan & Ng, Andrew & Potts, Christopher. (2011). Learning Word Vectors for Sentiment Analysis. 142-150.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.

[16] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543

[17] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In Advances in neural information processing systems, pages 3294–3302.

[18] Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. arXiv preprint arXiv:1803.02893.

[19] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In International conference on machine learning, pages 1188–1196.

[20] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365

[21] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364.

[22] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In Advances in Neural Information Processing Systems, pages 6294–6305.

[23] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428.

[24] Rish, Irina. (2001). An Empirical Study of the Naïve Bayes Classifier. IJCAI 2001 Work Empir Methods Artif Intell. 3.

[25] Rokach, Lior & Maimon, Oded. (2005). Decision Trees. 10.1007/0-387-25465-X_9.

[26] Cutler, Adele & Cutler, David & Stevens, John. (2011). Random Forests. 10.1007/978-1-4419-9326-7_5.

[27] Qader, Wisam & M. Ameen, Musa & Ahmed, Bilal. (2019). An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges. 200-204. 10.1109/IEC47844.2019.8950616.

[28] (2011). TF–IDF. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_83

[29] Mikolov, Tomas, Kai Chen, Gregory S. Corrado and Jeffrey Dean. “Efficient Estimation of Word Representations in Vector Space.” International Conference on Learning Representations (2013).

[30] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.