Two-step sentiment analysis (SA2)

Vincenzo Marco De Luca (224463)
University of Trento
Via Sommarive, 9, 38123 Povo,Trento TN
vincenzomarcodeluca@studenti.unitn.it

Abstract

Due to the spread of Web 2.0, the analysis of web opinion has became increasingly important, as it enables data scientist to extract useful information about the users. Companies immediately identified a relevant resource for their business, this is only one of the motivations behind the attention of research and development teams on Sentiment Analysis. This project explores some of Natural Language Processing techniques, that have been adapted to address Polarity Classification, objectivity detection, and proposes some adjustments to combine these two tasks and prevent the overfitting obstacle.

1 Introduction

Sentiment analysis (a.k.a. Polarity Detection) [16] is one of the most celebrated subtasks of Opinion Mining[2], it classifies a text into positive or negative based on the sentiment communicated by the user based on the words he writes. In 90s-00s, this task was addressed via traditional machine learning approaches, that after some preprocessing steps, extract features either manually or by means of feature selection methods and append on the top a classifier (e.g. Naive Bayes, SVM, Logistic Regression) to extract a probability, as showed in Figure 1.

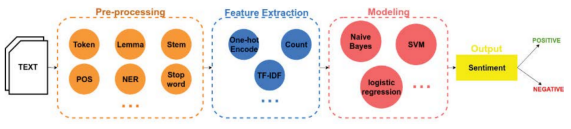


Figure 1: Traditional Machine Learning pipeline

After the backpropagation breakthrough, as in the other Data Science sub-fields, also Natural Language Processing (NLP) community moved its effort toward Deep Learning models, this process has been exponentially speeded up thanks to the era of Big Data, and hardware components able to compute much faster matrix computations. These deep Learning models directly integrates feature extraction at architecture level, as showed in Figure 2, thus they reduce the engineering effort, but also the model interpretability. Despite the interpretability difficulties, these models outperformed machine learning models even in Sentiment Analysis [16]. After the simple Fully Connected architectures (FCNN), they have been proposed many improvements that began with Recurrent Neural Networks (RNN)[29],pursued with Convolutional Neural Networks (CNN)[14], until Transformer invention[21]. Transformer-like architectures dramatically have changed the research direction, leading to bidirectional Encoder architectures (e.g. BERT [8], Roberta[23], XLNet [26], ELECTRA[13]).

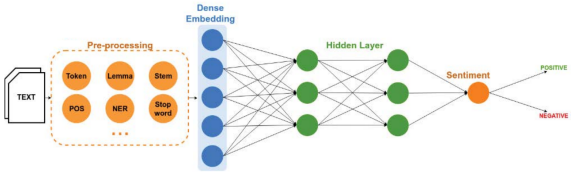


Figure 2: Traditional Deep Learning pipeline

In this project, the idea is to realize Sentiment Analysis as a two step procedure, in which a document is split in a sequence of sentences, and each sentence is classified as objective or subjective, if it is objective it will not be considered for the successive step that classifies the document polarity. The aim of objectivity removal is to prevent the network from considering objective sentences (that does not encode a user sentiment) during polarity classification. In order to solve this task, they have been explored the most celebrated NLP techniques, despite the non-trivial difficulty due to hardware limitation, the main effort has

been concentrated in a CNN-like architecture integrating multiple-parallel feature maps processing with variable kernel size to be that pooled overtime, Residual Connection, Squeeze and Excitation and some additional regularization techniques as it will be described in Section 3.

2 Proposed method

Given an input sentence, to realize tokenization and embedding, they have been experimented multiple approaches (Count Vectorizer, Tfidf Vectorizer and Hashing Vectorizer). As starting point to realize objectivity detection, it have been explored the Machine Learning approach based on the most celebrated classifiers (e.g. Support Vector Machines, Nu-Support Vector Machines, SGD, Decision tree, K-Nearest Neighbours, Random Forest). Since performances were not satisfying yet, the experiments have moved towards Deep Neural Networks, at the beginning the experiments with Recurrent Neural Network[25] with increasing order of complexity architectures, as described in Figure 6: Vanilla RNN[29], LSTM[19], [19], GRU[24], that provided increasing fair results, motivating us to proceeed trials only on GRU architecture and move to deeper architectures.

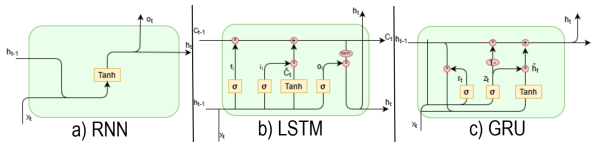


Figure 3: Main recurrent architectures

The successive trials have been realized on Convolutional Neural Networks[14].

The proposed architecture presents some novelties in this task, it takes inspiration from a paper written in 2005 [28], it processes in parallel the same embedding sequence, through 1D convolution layers realized by filter having different kernel size (in any case equal to odd numbers, moving by a step of 2) with stride set to 1.

The first contribution, with respect to the aforementioned paper, is to use only odd kernel sizes and apply multiple convolutional layers, not only in width, but also in depth. Because of the depth of these convolutional layers, it has been introduced also *parallel* Residual Connections[12], then it has been replaced by Dense Connections[3] for deeper architectures, and experimented Inception block, illustrated in Figure 7. Furthermore, the chan-

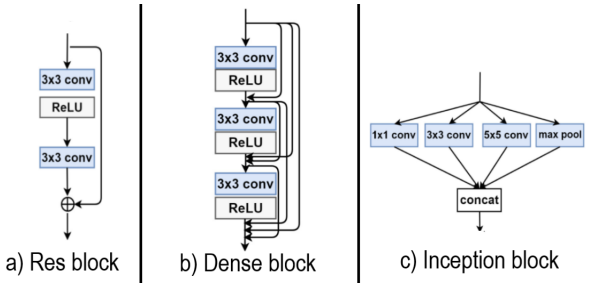


Figure 4: Baseline of feature extraction blocks

nel interdependencies has been improved at almost no computational cost by means of Squeeze and Excitation blocks[9], illustrated in Figure 8, in Convolutional Layers, the architecture has been named as ReSEntence. With respect to original architecture, it has been used AdamW[7] as optimizer, to integrate Weight Decay, then reduce generalization error and also guarantee faster convergence. As activation function, it has been embedded LeakyReLU[1] in the convolutional blocks and it has been experimented also other activation functions (RReLU, PrReLU, ELU, GELU, Swish[18],). Despite hard-parameter sharing, and Weight Decay, overfitting still took place, in order to prevent it we integrated some of the most celebrated regularization techniques for CNN[11], batch normalization (and some of its celebrated variants), hard dropout, aggressive Weight Decay, Learning rate on plateau, early stopping and parameter reduction. The need for reduction of parameters and, then reduce CNN depth, motivates to completely remove DenseBlocks. A last adjustment has been to reduce as much as possible the number of linear layers up to completely remove them. The architectures experimented for Sentiment Classification, illustrated in Figure 6, has been exactly the same even if the parameter tuning has been totally different, even they have been implemented and test in the source code multiple strategies to exploit NLTK-Vaeder framework. Once the Objective Neural Network (ONN) has been trained on a specific dataset, the knowledge

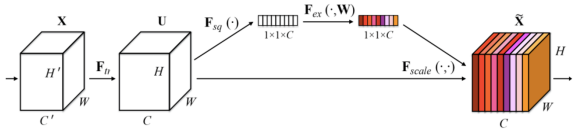


Figure 5: Squeeze and excitation block

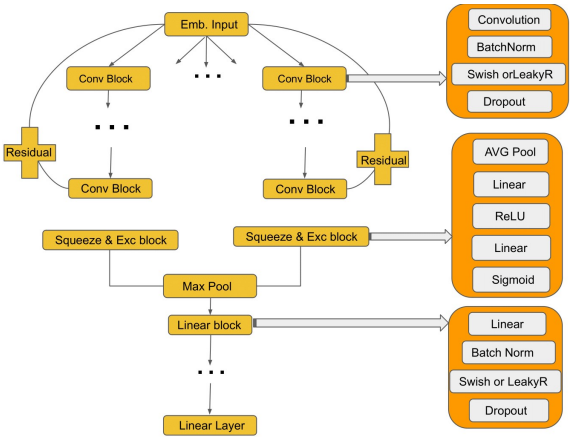
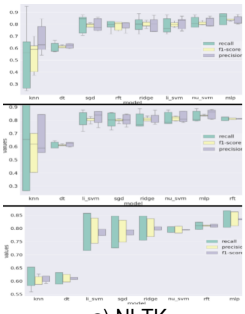


Figure 6: The proposed architecture: ReSEntence

acquired by the model has been re-used to detect subjective/objective sentences in the dataset used by Sentiment Neural Network (SNN), and filter out all the objective sentences from the document to be passed as input to the SNN, then the SNN will realize the final classification on the input dataset pre-processed by the ONN.

3 Experiments

They have been used three datasets for the experiments: the first two datasets are Movie Reviews datasets provided by NLTK and Pytorch while the third is the Cornell Movie Reviews dataset. The performances provided by different shallow classifiers and Vectorizer combinations have been resumed in Figure 7: on the left, they are reported the positive, negative precision, recall and F1 score grouped by Classifier and the three vectorizers under analysis measured on the NLTK dataset, whereas on the right there is the F1-score grouped by Classifier and the three vectorizers under analysis measured on the IMDB dataset. To



a) NLTK

Classifier	Count	Tfidf	Hash
nu svm	0.79	0.82	0.81
li svm	0.78	0.81	0.78
sgd	0.78	0.8	0.79
ridge	0.8	0.82	0.77
mlp	0.83	0.83	0.83
dt	0.61	0.61	0.61
knn	0.60	0.55	0.45
rft	0.81	0.8	0.74

b) IMDB

Figure 7: Shallow Model for objectivity detection, grouped by Vectorizer and Dataset

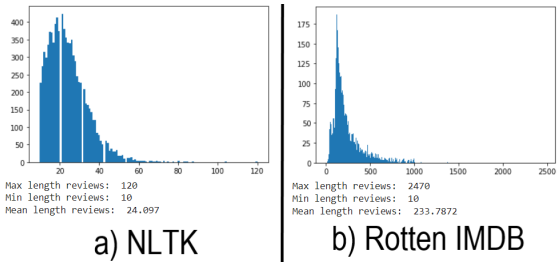


Figure 8: Histogram of sequence length of a) NLTK dataset and b) Rotten IMDB dataset

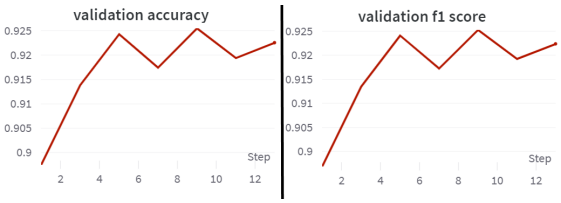


Figure 9: GRU performences in Objectivity detection on the Rotten IMBD dataset

move to deeper architecture, they are required preprocessing step, where the lengths of Objectivity and Sentiment sequences have been chosen based on the histograms of the length of sequences in the datasets, as illustrated in Figure 8. The best performances with a RNN-like architecture have been reached by a GRU that overfits after less than a dozen epochs as illustrated in Figure 9, but the model providing the best performances, showed in Figure 10 has been the ReSEntence, with parallel feature maps pooled over time as aforementioned in Section 3, by using two convolutional layers, hard dropout on CNN, just two fully connected layers, heavy weight decay (0.01), low learning rate (0.002) and Residual connection. Once trained the objectivity network, it has been possible to compare sentiment analysis with and without Objectivity removal. The first approach has been realized by means of NLTK-Vaeder, in Figure 11 are showed the confusion matrix obtained based on just four different algorithms, the

Figure 10: DenseSEntence performences in Objectivity detection on the Rotten IMBD dataset

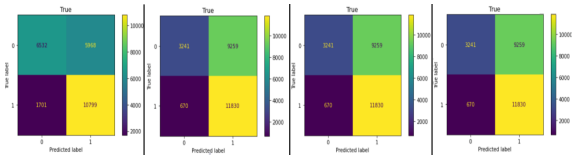


Figure 11: Confusion matrix in Sentiment analysis with NLTK-Vaeder

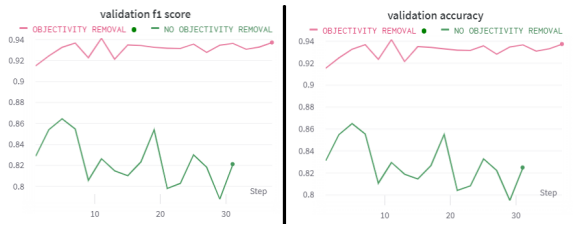


Figure 12: In red ReSEntence sentiment classification with objectivity removal (Wandb), in green ReSentence sentiment classification without objectivity removal(Wandb)

other algorithms proposed can be found in the associated notebook, , in first image from left, a strategy only based on sentiment analysis, in the second one a sentence-level strategy based on the count of positive and negative sentence, in the third one, a strategy based on the sum of negative and positive score at sentence level and in the forth one, a strategy based also on the compound component provided by NLTK framework. After some trials on the just discussed framework, it has been explored the ReSEntence approach for sentiment classification with and without objective sentence removal, as we can notice in figure 12, the sentiment classification with objective sentence removal outperform sentiment classification without sentence removal and is also less prone to overfitting. Further details about experiments, can be seen on the notebook (Tensorboard and file log) and the associated Wandb projects whose url can be found on the associated notebook.

4 Conclusion

Because of time reasons due to project deadline, there are multiple ideas that there have not yet been implemented and tested. First of all, it could be interesting a wider exploration of Movie Reviews datasets and of the vectorizer (e.g. Glove[17], Word2Vec[20]) to precisely compare the proposed DenSEntence to the other architectures proposed in literature. To realize a deeper CNN and simultaneously prevent overfitting, it could be possible to integrate Inception Module to DenseNet as proposed by Chen[27], and to regularize the NN: inject Adversarial Attack samples[6] to realize Adversarial Training[10] as proposed by Lin[4] and combine Sentiment and Objectivity tasks in a Multitask setting, that shares the dataset, and the backbone for both task while the final loss could be realized as a weighted sum of the losses from the two tasks based on learned weight parameters; in case of lack of enough large datasets, it could be possible to aggregate multiple datasets based on some Unsupervised Deep Domain Adaptation strategies. In case of robust hardware, it could be possible to combine advantages from GANBert[5] to the efficiency of ELECTRA[13] and the really recent and even more efficient version[22], to realize an efficient adversarial version of ELECTRA and adapt this approach to Italian Language models such as Alberto[15]

References

[1] Bing Xu et al. *Empirical Evaluation of Rectified Activations in Convolutional Network*. 2015. arXiv: 1505.00853 [cs.LG].

[2] Cortis Keith et al. "Over a decade of social opinion mining: a systematic review". In: *Artificial Intelligence Review* 54.7 (June 2021), pp. 4873–4965. ISSN: 1573-7462. DOI: 10.1007/s10462-021-10030-2. URL: http://dx.doi.org/10.1007/s10462-021-10030-2.

[3] Gao Huang et al. *Densely Connected Convolutional Networks*. 2018. arXiv: 1608.06993 [cs.CV].

[4] Gongqi Lin et al. *Commonsense knowledge adversarial dataset that challenges ELECTRA*. 2020. arXiv: 2010.13049 [cs.CL].

[5] Hoo-Chang Shin et al. *GANBERT: Generative Adversarial Networks with Bidirectional Encoder Representations from Transformers for MRI to PET synthesis*. 2020. arXiv: 2008.04393 [eess.IV].

[6] Ian J. Goodfellow et al. *Explaining and Harnessing Adversarial Examples*. 2015. arXiv: 1412.6572 [stat.ML].

[7] Ilya Loshchilov et al. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].

[8] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[9] Jie Hu et al. *Squeeze-and-Excitation Networks*. 2019. arXiv: 1709.01507 [cs.CV].

[10] Jin Yong Yoo et al. *Towards Improving Adversarial Training of NLP Models*. 2021. arXiv: 2109.00544 [cs.CL].

[11] João Paulo et al. "Avoiding Overfitting: A Survey on Regularization Methods for Convolutional Neural Networks". In: *ACM Computing Surveys* (Jan. 2021). ISSN: 1557-7341. DOI: 10.1145/3510413. URL: http://dx.doi.org/10.1145/3510413.

[12] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[13] Kevin Clark et al. *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*. 2020. arXiv: 2003.10555 [cs.CL].

[14] LeCun Y. et al. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

[15] Marco Polignano et al. "AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets". In: *CLiC-it*. 2019.

[16] Nhan Cach Dang et al. "Sentiment Analysis Based on Deep Learning: A Comparative Study". In: *CoRR* abs/2006.03541 (2020). arXiv: 2006.03541. URL: https://arxiv.org/abs/2006.03541.

[17] Pennington Jeffrey et al. "Glove: Global Vectors for Word Representation." In: *EMNLP*. Vol. 14. 2014, pp. 1532–1543.

[18] Prajit Ramachandran et al. *Searching for Activation Functions*. 2017. arXiv: 1710.05941 [cs.NE].

[19] Sepp Hochreiter et al. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[20] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].

[21] Vaswani Ashish et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[22] Yaru Hao et al. *Learning to Sample Replacements for ELECTRA Pre-Training*. 2021. arXiv: 2106.13715 [cs.CL].

[23] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[24] Yoshua Bengio et al. *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. 2014. arXiv: 1409.1259 [cs.CL].

[25] Yu Yong et al. "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures". In: *Neural Computation* 31.7 (July 2019), pp. 1235–1270. ISSN: 0899-7667. DOI: 10.1162/neco_a_01199. URL: https://doi.org/10.1162/neco%5C_a%5C_01199.

[26] Zhilin Yang et al. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. 2020. arXiv: 1906.08237 [cs.CL].

[27] Zhuo Chen et al. "DenseNet+Inception and Its Application for Electronic Transaction Fraud Detection". In: *2019 IEEE 21st International Conference on High Performance Computing and Communication*. 2019, pp. 2551–2558. DOI: 10.1109/HPCC/SmartCity/DSS.2019.00357.

[28] Yoon Kim. *Convolutional Neural Networks for Sentence Classification*. 2014. arXiv: 1408.5882 [cs.CL].

[29] Elman Jeffrey L. "Finding structure in time". In: *Cognitive Science* 14 (1990), pp. 179–211.