Quest Journals
Journal of Software Engineering and Simulation
Volume 10 ~ Issue 5 (2024) pp: 12-15
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Twitter Sentiment Analysis with CNNs

Garima Chandore Anusha Jain Aryan GuptaYash Palod
Assistant ProfessorAssistant ProfessorDept. Computer Science Engineering Dept. Computer Science Engineering
Medi-Caps University Medi-Caps UniversityMedi-Caps University Medi-CapsUniversity
Indore, India Indore, IndiaIndore, India Indore, India

Abstract
This research delves into our engagement in the SemEval-2017 Twitter sentiment analysis challenge, where our objective was to improve sentiment classification for tweets using deep learning models and advanced training techniques. We developed an ensemble model comprising 10 CNNs with diverse hyperparameters and pre-training strategies, which led to our top-ranking performance across all English subtasks. Moving forward, future research could explore seamless integration of CNNs and investigate models similar to those proposed by Stojanovski et al. (2016). Additionally, analyzing the impact of varying amounts of unlabeled and distant data on model performance could provide valuable insights for further enhancing sentiment analysis systems.

Keywords-*Sentiment Analysis, Twitter, Convolutional Neural Networks, Social Media, Machine Learning, Natural Language Processing*

R*eceived 29 Apr., 2024;* ***R****evised 06 May, 2024;* ***A****ccepted 08 May, 2024 © The author(s) 2024.*
P*ublished with open access at* ***www.questjournals.org***

I. Introduction

In contemporary data science and natural language processing (NLP), assessing the emotional tone or attitude conveyed in tweets indispensable. Its widespread appeal lies in its simplicity and the potential for favorable outcomes through basic techniques such as tallying positive and negative word frequencies. This field holds vast practical implications, ranging from monitoring high-profile events like the Oscars and presidential debates to generating trading signals through the analysis of tweets related to publicly traded companies. Achieving high accuracy is often imperative for these real-world applications, driving initiatives like the SemEval-2017 Twitter challenge to encourage further exploration in this area. As outlined by Rosenthal et al. (2017), the competition encompasses five distinct subtasks, including distributional estimation, ordinal classification, and standard classification.

Sentiment analysis is only one of the NLP tasks in which deep learning approaches have greatly surpassed conventional methods in recent years. Previous editions competitors for SemEval's Twitter sentiment analysis have alreadyshowcased the superiority of deep learning over alternative approaches. Among the favored deep learning models for sentiment analysis are Convolutional Neural Networks (CNNs). Therefore, in our endeavor to create an advanced Twitter sentiment classifier, we integrate the CNNs model into our system.

This manuscript is organized as follows:Sectiontwogives the architecture of the CNNmodel utilized in our system. Section three elaborates on the three training phases implemented. Section 4 discusses the various strategies employed enhance the system for every individual task in the project. .Last, Section five presents system's performance, while Section 6 summarizes our key findings

II. Literature Review

Twitter sentiment analysis has become a significant area of research due to the massive amount of user-generated content on the platform and the need to understand public opinion. Convolutional Neural Networks (CNNs) have emerged as effective tools for sentiment analysis tasks due to their ability to capture local dependencies in textual data. This literature review aims to explore and analyze previous studies focusing on Pang et al. introduced one of the earliest works on sentiment analysis, utilizing machine learning techniques such as Naive Bayes, Support Vector Machines (SVM), and Maximum Entropy models. This work laid the foundation for subsequent studies in sentiment analysis.

Kimproposed the use of CNNs for sentence classification tasks, achieving state-of-the-art performance on various benchmark datasets. CNNs were shown to effectively capture n-gram features and hierarchical structures in text data.Zhang et al. presented a deep CNN architecture for sentiment analysis, demonstrating its effectiveness on a variety of datasets. They employed multiple convolutional layers followed by max-pooling and achieved competitive results without the need for handcrafted features.Severyn and Moschitti introduced a CNN-based model for sentiment analysis in Twitter messages. They utilized pre-trained word embeddings and multiple convolutional layers with different filter widths to capture varying levels of textual features.dos Santos and Gatti proposed a CNN architecture incorporating dynamic k-max pooling for sentiment analysis tasks. Their model achieved competitive results on various benchmark datasets while being computationally efficient.Pak and Paroubek conducted one of the early studies on sentiment analysis specifically focusing on Twitter data. They explored lexicon-based approaches and machine learning techniques to classify tweets into positive, negative, and neutral sentiments.Go et al.introduced the Stanford Twitter Sentiment corpus, a widely used dataset for sentiment analysis research on Twitter. Their work provided a labeled dataset of tweets for training and evaluating sentiment analysis models.Bermingham and Smeatoninvestigated sentiment analysis on Twitter during specific events, such as elections and natural disasters. They highlighted the challenges of sentiment analysis in real-time and dynamic environments.

Twitter data often contain misspellings, slang, and grammatical errors, posing challenges for sentiment analysis models.Models trained on generic sentiment analysis datasets may not perform well when applied to domain-specific Twitter data. Future research could focus on detecting nuanced sentiments such as sarcasm and irony in Twitter data using advanced deep learning techniques.

III. Description of the System

A. CNN's

The architecture of the CNN we utilized closely resembles Kim's CNN model from 2014. Our model, a scaled-down version, processes input tweets by tokenizing them into words and then mapping each word to a word vector representation (word embedding). This process leads to the creation of a matrix sized s x g, where s signifies count of words within the tweets , and g denotes the selected dimensionality of the embedding space (in our instance, g = 200).To ensure uniformity in matrix dimensions, we employ a zero-padding strategy inspired by Kim (2014), setting the matrix size to Rs' x g, with s' = 80.

Multiple convolution operations of varying sizes are then applied to this matrix. In each convolution, a filtering matrix denoted as w is employed. of size Rh x d, where h denotes the convolution size (number of words spanned). By employing different filter sizes, such as, , or, and utilizing 200 filtering matrices for each size, we can focus on different regions of the tweets.

Following the convolutions, a max-pooling operation is performed on each convolution result. This operation, represented by In this equation, ci is computed as the result of applying a function f to the sum of the convolution operation on the input matrix X with the filter wj,k, along with the bias term b. chosen non-linear function (in our case, the ReLU function), extracts key features independently of their position in the tweet. The output cmax is obtained by taking the maximum value from ci. This process effectively captures essential n-grams in the embedding space, enhancing sentence classification capabilities.

In the last stage, all maximum cmax values extracted from each filter are consolidated into a vector denoted as cmax, which possesses a dimensionality of rm, where m stands for the total number of filters This resulting vector then undergoes processing through a compact fully connected hidden layers comprising 30 nodes, followed by a softmax layer for computing classification probabilities. To mitigate the risk of overfitting, dropout layers are introduced during training, with a dropout probability of 50%, following both the max-pooling and fully connected layers

IV. Training

We used a comprehensive approach to train our models, combining a significant number of unlabeled tweets with human-labeled data. Thirty,849 tweets for subtasks C and E, 49,693 human-labeled tweets for subtask A, and 18948 tweets for subtasks made up our dataset. We also included 100 million English tweets without labels that we pulled from the Twitter streaming API. Using a remote dataset, we were able to extract 5 million positive and 5 million negative tweets from this unlabeled dataset. creation method based on emoticon associations, following the approach by Go et al. (2009). These three datasets - labeled, unlabeled, and distant - were separately employed in distinct training stages, aligning closely with methodologies seen in prior works lenabling a comprehensive and diverse training strategy to enhance the robustness and efficacy of our models.

A.Data Preparation

Before inputting tweets into any training phase, undergo preprocessing via the following steps:

- URL Replacement: Any URL's present in tweets are substituted with the token "<url>" to standardize their representation..
- Letter Repetition Reduction: To streamline text, any letter repeated more than twice consecutively is truncated to two repetitions (e.g., "sooooo" becomes "soo").
- Lowercasing: All tweets are converted to lowercase, facilitating uniform processing and analysis during subsequent training stages.

B. Unsupervised Training

The pre-training process for word embeddings using 100 million unlabeled tweets involved experimenting with three unsupervised learning algorithms: Google's Word2vec, Facebook's FastText, and Stanford's GloVe. Word2vec focuses on learning fasttext generates word vector representations by predicting the context words surrounding a given input word. This approach is akin to Word2vec. incorporates subword information in its prediction model. In contrast, GloVerepresents a model architecture that relies on global word-word co-occurrence statistics. The implementation of these algorithms utilized the default settings and code provided by the respective authors to generate the word embeddings essential for subsequent use in the CNN's.

C. Distant Learning

Based on provided sources, the process of enhancing word embeddings with sentiment polarity information involves a multi-step approach. Initially, embeddings learned in an unsupervised phase lack sentiment polarity distinctions, as positive and negative words share similar contexts. To address this, a fine-tuning step via distant training is employed after unsupervised training. In this phase, a CNN model is utilized, initialized with the unsupervised embeddings. The CNN is trained on a distant dataset to differentiate noisy positive tweets from noisy negative tweets. Initially, the embeddings are frozen to minimize drastic changes, followed by six epochs of training with unfrozen embeddings. This process ensures that words with contrasting sentiment polarities are distinctly positioned in the embedding space, enhancing model's ability to capture sentiment nuances effectively.

D. Supervised Learning

In the last training phase, SemEval-2017's human-labeled data is used. The CNN models' embeddings are first initialized using the refined embeddings from the remote training phase, after which they are frozen for the first five epochs. Then, training is carried out for five more epochs using unfrozen embeddings and a ten-fold lower learning rate. To deal with the unbalanced dataset, the cross-entropy loss function was selected and weighted by considering the rarity of occurrences within the true classes. The Adam optimizer, which is run in TensorFlow on a GeForce GTX Titan X GPU with an initial learning rate of 0.001, is used in the optimization process. By integrating 10 CNNs and 10 LSTMs via soft voting, an ensemble technique is used to improve accuracy and decrease variation.The ensemble models include distinct embedding pre-training algorithms (Word2vec or FastText), different sets of filter sizes ([1, 2, 3], [3, 4, 5] or [5, 6, 7]),employing a range of diverse random weights at the initiation of the training phase, encompassing epochs that range from 4 to 20

V. Subtask-Specific Strategies

Incorporating the models and training methodology outlined in Section 2 and Section 3 across all five subtasks involves specific adaptations to addressunique requirements. It's worth noting that the dimension of the output varies contingent upon the specific subtask: it is 3 subtask , 2 for subtasks B and D, and 5 for subtasks C and E. For quantification tasks (specifically D and E), a probability averaging technique inspired by Bella et al. to transform the probabilities generated as output into a distribution reflecting sentiment variations.

Additionally, for subtasks related to tweet themes two additional steps are implemented to enhance accuracy during cross-validation. Firstly, if any topic-related words are absent in the tweet, these missing words are appended to the end of the tweet during preprocessing. Secondly, a specialized embedding space of dimension 5 is concatenated with the regular word embeddings. This additional space contains only 2 possible vectors: one indicating the current word's relevance to the topic and the other denoting its lack of association with the topic. These tailored adjustments optimize model performance and accuracy across the diverse subtasks, ensuring effective sentiment analysis and classification.

VI. Result

The evaluation of model performance on the historical Twitter test sets from 2013 to 2016, focusing primarily on Task A, reveals insightful findings. The assessment metric utilized is average F1 score for the positive and negative classes, aligning with past competition standards. Notably, the GloVe unsupervised algorithm yields lower scores compared to FastText and Word2vec, leading to its exclusion from the ensemble model. It is evident that incorporating class weights and a distant training stage significantly enhances performance, as observed by the notable score improvements. While other models exhibit similar performance levels, the ensemble model emerges as the top performer, surpassing individual models. The ensemble's success stems from the diverse outputs of individual models, which when combined, synergistically boost overall performance. Calculating Pearson correlation coefficients between model output probabilities further highlights the independence of predictions within the ensemble. Models from distinct learning approaches and varied unsupervised learning methods contribute to this

System	2013	2014	2015	2016
Logisticregression	0.625	0.625	0.584	0.553
CNN(convolutionsize=[3,4,5],word2vec)	0.710	0.724	**0.682**	0.642
CNN(convolutionsize=[3,4,5],fasttext)	0.721	0.732	0.663	0.641
CNN(convolutionsize=[3,4,5]),glove)	0.710	0.715	0.661	0.631
CNN(convolutionsize=[1,2,3],word2vec)	0.713	0.733	0.672	0.641
CNN(convolutionsize=[5,6,7],word2vec)	0.711	0.731	0.675	0.642
CNN(word2vec,convolutionsize=[3,4,5])	0.683	0.675	0.654	0.642
CNN(word2vec,convolutionsize=[3,4,5])	0.694	0.713	0.661	0.633
CNN(word2vec,convolutionsize=[3,4,5])	0.710	0.720	0.682	0.642
Ensemblemodel	0.722	**0.742**	0.678	**0.645**
Previousbesthistoricalscores	**0.727**	0.741	0.670	0.632

ensemble's effectiveness in achieving superior results.

The study on ensemble learning-based models for Twitter spam detection revealed promising performances of the models built. Ensemble models, as discussed in the sources, involve combining decisions from multiple models to improve overall performance by overcoming issues like noise, bias, and variance. These models use techniques like Max Voting, Averaging, and Weighted Average to aggregate predictions for classification and regression problems. In a specific study on Omicron Tweet Sentiment Analysis, ensemble meta-classifiers showed strong prediction results compared to single classifier models, with the voting classifier achieving 85.33% accuracy and the stacking classifier reaching 88.06% performance accuracy.

VII. Conclusions

This study details our participation in the SemEval Twitter sentiment analysis competition, where we aimed to enhance sentiment classification for tweets by leveraging deep learning models and advanced training methodologies. Our approach culminated in an ensemble model comprising 10 CNNs with distinct hyperparameters and pre-training strategies, leading to our top-ranking performance across all English subtasks.

Moving forward, future investigations could delve into integrating CNNs seamlessly, potentiallyexploringmodels akin to those proposed by Stojanovski et al. (2016). Additionally, exploring the impact of varying amounts of unlabeled and distant data on model performance could provide valuable insights for further enhancing sentiment analysis systems.

Refrences

[1]. The paper authored by Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio in 2014 presents a neural machine translation approach that involves simultaneous learning of alignment and translation. The work is documented in the arXiv preprint with the identifier arXiv:1409.0473

[2]. In their 2016 arXiv preprint, Bojanowski, Grave, Joulin, and Mikolov delve into the enhancement of word vectors through the integration of subword information. This seminal work explores novel approaches to enriching language representations, providing valuable insights for natural language processing tasks.

[3]. In the proceedings of EMNLP in 2014, Danqi Chen and Christopher D. Manning introduced a rapid and precise dependency parser leveraging neural networks. Their research, spanning pages 740–750 of the conference, outlines a groundbreaking approach to dependency parsing with neural network techniques.

[4]. The paper titled "BB twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs" explores the application of convolutional neural networks (CNNs) and long short-term memory networks (LSTMs) for sentiment analysis on Twitter data, as presented at SemEval-2017 Task 4.