Text Mining Final Assignment: Sentiment Analysis in Twitter using Biattentive Classification Network*

Zhipei Qin[1 [s3977226]] and Peiwen Xing[2 [s3838501]]

Leiden University, Rapenburg 70, 2311 EZ Leiden, the Netherlands

Abstract. "Sentiment Analysis in Twitter" is a widely recognized task that attracts attention every year. This paper introduces a Biattentive Classification Network (BCN), which is designed for the SemEval-2017 Task 4 (Subtask A: Overall Sentiment of a Tweet). The network employs a biattention mechanism to process the encoder's output, generating a focused vector representation of text, and built upon word embeddings pre-trained by GloVe or fastText. We have established a whole process to load and preprocess Twitter data, making it conducive for subsequent BCN network processing. We experiment with adding extra Bi-LSTM layers to the BCN encoder to increase the networks' complexity. In the experiement, we select our best model (GloVe+BCN with 2Bi-LSTMs) among 6 model candidates based on average recall, and this best model is further compared with two state-of-the-art models: LSTMs+CNNs in the BB_twtr system and Deep Bi-LSTM+attention in DataStories system. The results show that our BCN model shows slight improvements over these implemented models in the task.

Keywords: Sentiment Analysis · Twitter · Biattentive Classification Network · GloVe · fastText · Bi-LSTM.

1 Introduction

Sentiment analysis in textual content is a significant area of research, increasingly attracting attention in the realms of language processing and social science, particularly with the rise of social media platforms like Twitter. This analytical task typically requires identifying if a text conveys a POSITIVE, NEGATIVE, or NEUTRAL sentiment. This sentiment could be overarching or related to a specific subject, such as an individual, a product, or an event. In this project, we mainly focus on the sentiment analysis task of Twitter. Performing sentiment analysis on Twitter is particularly challenging due to the informal and unique writing styles often found on the platform. On this social media platform, user interactions primarily occur through brief messages known as tweets. It is estimated by the company that more than 500 million tweets are dispatched daily. Although these tweets are limited to a maximum of 140 characters, they are a

* Supported by Leiden University.

valuable source of data, as they often contain users' personal thoughts, opinions, and emotional expressions.

SemEval (Semantic Evaluation) is an international competition focusing on various tasks in Natural Language Processing (NLP) and semantic analysis, particularly in areas such as sentiment analysis, word sense disambiguation, and semantic relation identification. Twitter sentiment analysis has consistently been a pivotal task for SemEval, starting from 2013. This task is registered as **SemEval Task 4**. The 2015 edition of the task expanded to include sentiments regarding particular topics [Rosenthal et al.(2019b)], while in 2016, the challenge evolved to encompass tweet quantification and a detailed five-level ordinal classification [Nakov et al.(2019)].

SemEval-2017 Task 4 encompasses both tweet classification and tweet quantification tasks, utilizing 2-point and 5-point scales. The details of the task result are discussed in [Rosenthal et al.(2019a)]. In this project, we only deal with the classification subtask in English with no topic given (i.e. subtask A), which focuses on determining the overall sentiment expressed in a specific tweet. This subtask can be defined as:
Given a tweet, determine whether a tweet conveys a POSITIVE, NEGATIVE, or NEUTRAL sentiment.

In the data preprocessing section of our project, we created a process for processing Twitter data. This process mainly includes data reading, preprocessing, word segmentation, removing stop words and other irrelevant characters, and converting the processed data into a format suitable for machine learning models.

We further implement BCN (Biatentive Classification Network) discussed in [McCann et al.(2018)] as the main deep learning model architecture for this sentiment analysis task. BCN integrates and compares information from different parts of a text sequence by using an attention mechanism. This mechanism allows the model to focus on the most relevant parts of the text, thereby improving the understanding of text content and classification accuracy. This network accommodates both single-sentence and two-sentence classification tasks, treating single-sentence inputs as duplicated sequences. The network architecture includes the following parts: Word Embedding Layer, Feedforward Layer, Encoder Layer, Bi-Attention Mechanism Layer, Integrator layer, Pooling layer and output layer. A detailed introduction to BCN will be given in Chapter 4.

In the experiment, we employ two popular pre-trained methods, GloVe (Global Vectors for Word Representation) and fastText, to obtain the English word representation. GloVe focuses on global word co-occurrence statistics, while fastText uses CBOW with position-weights to generate word vectors, therefore, it mainly

uses local word context.

In SemEval-2017 Task 4, models using Convolutional Neural Networks (CNNs) and Long Short Term Memory (LSTMs) networks [Cliche(2017)] and Deep Bi-LSTM+attention [Baziotis et al.(2017)] are the two state-of-the-art models for this task. Our models compare with these two models based on three performance indicators: F1-score (*f1*), accuracy(*acc*) and average recall (*AvgRec*).

2 Related Work

Many approaches have focused on sentiment analysis in Twitter. Traditional approaches like hand-crafted features and sentiment lexicons have been used. For example, Edilson A. et al. [au2 et al.(2017)] introduce a multi-view ensemble method for text polarity classification, integrating bag-of-words and word embeddings with Support Vector Machines (SVM) and Logistic Regression. By Dmitry Ignatov et al. [Ignatov and Ignatov(2017)], a distinctive architecture known as a Decision Stream is introduced. This method focuses on merging nodes from differing branches during the learning process, based on their similarity assessed by two-sample test statistics. This approach leads to the formation of a complex, directed acyclic graph with decision rules, spanning potentially hundreds of levels, instead of a traditional tree structure.

However, Deep learning, particularly CNNs and LSTMs, has outperformed traditional methods in sentiment analysis. Mathieu Cliche[Cliche(2017)] used an ensemble of 10 CNNs and 10 LSTMs with different hyper-parameters and different pre-training strategies in the SemEval-2017 Twitter sentiment analysis competition. They utilized a significant amount of unlabeled data for pre-training word embeddings and fine-tuned them with distant supervision. The ensemble of CNNs and LSTMs achieved first place in all five English subtasks in the SemEval-2017 competition, showcasing their success in deep learning models for sentiment analysis. Christos Baziotis et al. [Baziotis et al.(2017)] introduces two deep-learning systems for SemEval-2017 Task 4, aiming to improve message-level and topic-based sentiment analysis. The first model uses a 2-layer Bidirectional LSTM with attention, while the second employs a Siamese Bidirectional LSTM with context-aware attention. In the task of "Overall Sentiment in a Tweet," the model achieved second place.

In this project, rather than using CNNs or simple LSTMs, we adopt the model architecture proposed by [McCann et al.(2018)]: Biattentive Classification Network (BCN). The described Biattentive classification network in [McCann et al.(2018)] is used to evaluate the transferability of CoVe to various tasks. It employs a feedforward network with ReLU activation and a biLSTM encoder for generating task-specific representations of each input. Bi-attention allows each sequence to condition on the other, further integrated by a biLSTM. Finally, a maxout network with pooled features determines the class distribution. We anticipate

that the BCN model we adopted will surpass the current leading models in performance on SemEval-2017 Task 4 (subtask A). In particular, we hope that BCN will achieve the best results on *AvgRec* scores, as it is the primary measure mentioned in [Rosenthal et al.(2019a)].

3 Data

The dataset for Task 4, Subtask A, consists of a series of tweets, each accompanied by sentiment labels categorized as "positive," "negative," or "neutral." Each tweet is assigned a unique ID number. The content of the tweets reflects users' expressions of praise, criticism, or neutrality towards various entities. In this paper, we utilize "twitter-2016train-A" as the training set, "twitter-2016test-A" as the test set, "twitter-2016dev-A" as the development set, and "twitter-2016devtest-A" as the development test set. The distribution of labels in the dataset is shown in the table below:

Table 1. Statistics about the datasets of the task.

Dataset	Positive	Neutral	Negtive	Total
Train	3094	2043	863	6000
Test	7059	10342	3231	20632
Dev	843	765	391	1999
Devtest	994	681	325	2000

4 Methods

4.1 Data Pre-processing

Prior to feature extraction, the tweets underwent a preprocessing phase. Initially, the task involves reading Twitter data from text files, including recognizing tweet IDs, the content of the tweets and their associated sentiment labels. Subsequently, these tweets are cleaned, which includes removing useless characters, and HTML entities, and standardizing the text. Following this, a tokenization tool is used to break the text down into words or tokens, while eliminating stop words and other irrelevant information such as punctuation, URLs, mentions, and hashtags. Finally, the cleaned and preprocessed text is converted into a format that can be accepted by machine learning models, to be used for further text analysis or sentiment classification tasks.

4.2 Word embeddings

We used two kinds of pre-trained word embeddings. One word embedding is generated with the GloVe model [Pennington et al.(2014)] and trained on the Common Crawl dataset with 840 billion cased tokens. The embedding vectors are 300 dimensions. The other pre-trained English word embeddings are developed using fastText[Bojanowski et al.(2016)], a library created by Facebook's AI Research lab for learning word embeddings and text classification. These embeddings were trained on data from Common Crawl and Wikipedia. The training utilized the CBOW model with position-weights, and the embeddings are 300-dimensional. They include character n-grams of length 5, a window of size 5, and 10 negatives.

4.3 Biattentive Classification Network Model

Our Biattentive Classification Network (BCN) employs a Bi-directional LSTM (BiLSTM) encoder, with a bi-attention mechanism to pinpoint the most significant words, and is registered as "*my_bcn*" within the AllenNLP [Gardner et al.(2018)] framework.

The main components and workflow of BCN are:

1. **Word Embedding Layer**: BCN builds a text embedder to convert the words in the input text into vector representations through the embedding layer. In our BCN model, we combine the pre-trained ELMo [Peters et al.(2018)] representation with the input or Integrate the encoder output connection, which dynamically adjusts the embedding representation based on the context of the word in the text. The use of ELMo can provide rich, context-sensitive word embeddings. Before being passed to the feedforward network, the embedding vector goes through a dropout layer, whose parameters we set to 0.4.

2. **Feedforward Layer**: The embedding vectors are initially processed through a feedforward neural network. In our model, the input dimension is 1324, and a hidden layer with dimension 300 is set. We use ReLU as the activation function and set the dropout ratio to 0.2.

3. **Encoder Layer**: Send the processed vector to the sequence encoder to encode the entire sequence. We use bidirectional LSTM (BiLSTM) in order to get word annotations that summarize the information from both directions. We first try one layer of BiLSTMs and then we stack two layers of BiLSTMs in order to learn more abstract features.

4. **Bi-Attention Mechanism Layer**: The model uses the bi-attention mechanism to process the output of the encoder and obtain a vector representation that focuses on the text. By calculating attention logits, the correlation of each word with all other words in the sequence is calculated, forming an attention

logits matrix. After that, use a mask to cover up those parts that are not real text components, and use the softmax function to normalize the attention logit matrix. Finally, the encoded-word vectors are weighted and summed using the calculated attention weights. This means that the representation of each word contains not only its own information but also its relationship to other parts of the text.

5. **Integrator layer**: Combine the Bi-Attention processed text representation with the previous encoder output, and then further integrate it through another biLSTM encoder (named integrator).

6. **Pooling layer**: The output from the integrated encoder undergoes max pooling, min pooling, mean pooling, and self-attention pooling to generate the final text representation.

7. **Output layer**: The final text representation is generated through a maximization network (Maxout) or feedforward layer (FeedForward) to generate classification predictions.

5 Experiments and Results

In this section, we conducted a comparative analysis of different word embedding models and various numbers of BiLSTM layers. We focused on word embedding models such as GloVe and model in fastText (i.e. CBOW with position-weights), each having distinct characteristics and performance in semantic representation. Simultaneously, we investigated the impact of the number of BiLSTM layers on model performance, exploring the differences between single-layer BiLSTM and multi-layer BiLSTM. Finally, we compared our best-performing model with two state-of-the-art models mentioned in Chapter 2. The experimental results are shown below:

Word Embedding	BiLSTM Layers	*AvgRec*	*F1*	*Acc*
GloVe	1	0.683	**0.682**	0.687
	2	**0.686**	0.677	0.687
	3	0.679	0.676	**0.689**
fastText	1	0.679	0.676	0.686
	2	0.683	0.675	0.684
	3	0.679	0.676	**0.689**

Table 2. Comparison of different word embedding models and varying numbers of BiLSTM layers.

Model	*AvgRec*	*F1*	*Acc*
GloVe + 2BiLSTMs	**0.686**	0.677	**0.687**
DataStories	0.681	0.677	0.651
BB_twtr	0.681	**0.685**	0.658

Table 3. Comparison between Glove+2BiLSTMs and two state-of-the-art models.

Experimental setup We conducted the training of our models under the AllenNLP framework [Gardner et al.(2018)]. We trained our neural networks on a NVIDIA GeForce RTX 3090.

6 Discussion

We experimented with six model combinations using GloVe and fastText embeddings, each paired with one, two, or three BiLSTM encoders. The results across these six models show no significant differences. GloVe with two BiLSTM encoders achieved the best *AvgRec* score, while GloVe with a single BiLSTM encoder scored highest in F1. The accuracy of all six models was nearly identical. Overall, using GloVe and fastText yielded similar results, and adding extra LSTM layers in the BCN's encoder did not significantly enhance performance. As *AvgRec* is the primary metric highlighted in [Rosenthal et al.(2019a)], we selected GloVe+2BiLSTMs as our top-performing model, for comparison with the two best existing models, LSTMs+CNNs (BB_twtr) and Deep BiLSTM+attention (DataStories). Our model outperformed these two in average recall and accuracy.

7 Conclusion

In our paper, we introduce the Biattentive Classification Network (BCN) for sentiment analysis of short texts, specifically designed for the SemEval-2017 Task 4 "Sentiment Analysis in Twitter" (subtask A). We start with basic preprocessing of the original tweet data, including tokenization, removing stopwords, quotes, escape characters, decoding HTML entities, and normalizing spaces, etc. We then explore two word embedding models, GloVe and fastText, to enhance the model's effectiveness. Pre-trained ELMo representations are also used in our BCN's input. Our core idea is to implement the Biattentive Classification Network model and experiment with adjustments to its encoder structure. Our models achieved outstanding results in the task, surpassing two top-performing models under the *AvgRec* metric. Future approaches of interest include designing the BCN model for Topic-Based Classification and Tweet Quantification tasks, as well as investigating BCN's transferability across different languages.

8 Contributions of the team members

Zhipei Qin: Abstract, Introduction, Related Work, Methods, Experiments and Results, Discussion, Conclusion
Peiwen Xing: Related Work, Data, Experiments and Results, Discussion

Acknowledgements We would like to thank the other teammate for his contributions to this coursework, and professor Suzan for the breadth of knowledge and deep insights provided in this text mining topic.

References

[au2 et al.(2017)] Edilson A. Corrêa Jr. au2, Vanessa Queiroz Marinho, and Leandro Borges dos Santos. 2017. NILC-USP at SemEval-2017 Task 4: A Multi-view Ensemble for Twitter Sentiment Analysis. arXiv:1704.02263 [cs.CL]

[Baziotis et al.(2017)] Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens (Eds.). Association for Computational Linguistics, Vancouver, Canada, 747–754. `https://doi.org/10.18653/v1/S17-2126`

[Bojanowski et al.(2016)] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606* (2016).

[Cliche(2017)] Mathieu Cliche. 2017. $\mathrm{BB}_t wtrat SemEval - 2017 Task4 : TwitterSentimentAnalysiswithCNNsandLSTMs. arXiv : 1704.06125 [cs.CL]$

[Gardner et al.(2018)] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. arXiv:1803.07640 [cs.CL]

[Ignatov and Ignatov(2017)] Dmitry Ignatov and Andrey Ignatov. 2017. Decision Stream: Cultivating Deep Decision Trees. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. 905–912. `https://doi.org/10.1109/ICTAI.2017.00140`

[McCann et al.(2018)] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2018. Learned in Translation: Contextualized Word Vectors. arXiv:1708.00107 [cs.CL]

[Nakov et al.(2019)] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2019. SemEval-2016 Task 4: Sentiment Analysis in Twitter. arXiv:1912.01973 [cs.CL]

[Pennington et al.(2014)] 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)*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). Association for Computational Linguistics, Doha, Qatar, 1532–1543. `https://doi.org/10.3115/v1/D14-1162`

[Peters et al.(2018)] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. arXiv:1802.05365 [cs.CL]

[Rosenthal et al.(2019a)] Sara Rosenthal, Noura Farra, and Preslav Nakov. 2019a. SemEval-2017 Task 4: Sentiment Analysis in Twitter. arXiv:1912.00741 [cs.CL]

[Rosenthal et al.(2019b)] Sara Rosenthal, Saif M Mohammad, Preslav Nakov, Alan Ritter, Svetlana Kiritchenko, and Veselin Stoyanov. 2019b. SemEval-2015 Task 10: Sentiment Analysis in Twitter. arXiv:1912.02387 [cs.CL]