Exploring the Attention Mechanism in Deep Models: A Case Study on Sentiment Analysis

Martina Toshevska(✉) and Slobodan Kalajdziski

Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University,
Skopje, Macedonia
{martina.toshevska,slobodan.kalajdziski}@finki.ukim.mk

Abstract. Interpreting what a deep learning model has learned is a challenging task. In this paper, we present a deep learning architecture relying upon an attention mechanism. The main focus is put on the exploratory evaluation of attention-based deep learning models on lexicons of affective words, and examination whether the word valence is the most significant information or not. Obtained evaluation results lead to a conclusion that word valences do play a significant role in sentiment analysis, but possibly models rely upon other dimensions perhaps not distinguishable by humans.

Keywords: Deep learning · Attention mechanism · Sentiment analysis · Sentiment lexicons · Word valence

1 Introduction

Analyzing people's attitude (opinion, sentiment, emotions) has received significant attention from the researchers. Variety of models for identifying sentiment (emotion, opinion) are already created with both machine and deep learning techniques.

Machine learning techniques rely mostly on feature engineering that sometimes depends on external resources. Lexicons of affective words and phrases are frequently used for extracting features which are later fed into a machine learning model [7,8,11]. Therefore, we can hypothesize that affective words and their associated information are an essential indicator of the sentiment. One type of information provided in lexical resources is valence. It refers to whether a word is considered positive or negative.

On the other side, deep learning models mostly rely only on the words present in the text [1,5,18]. Nevertheless, recent trends about incorporating lexical information into the deep model attract the researchers [16,17]. Deep learning models are built upon most common architectures recommended in the field of natural language processing - recurrent neural networks (RNNs) and convolutional neural networks (CNNs). Enhancing the model with an attention mechanism leads

S. Gievska and G. Madjarov (Eds.): ICT Innovations 2019, CCIS 1110, pp. 202–211, 2019.
https://doi.org/10.1007/978-3-030-33110-8_17

to promising performance improvement. The attention mechanism enables the model to focus on the part of the input.

However, what does the model, in fact, learn? Deep learning models achieve high accuracy at the expense of their interpretability since these models are often treated as black-box models. Analyzing deep learning models is often a challenging task. In what follows we highlight the findings of our experimental evaluation of various deep models. We examine their learned weights and to what extent they correlate to the information confirmed to provide reliable features for classical machine learning models. The rest of the paper is organized as follows. Section 2 describes in details the steps of our research. Discussion of experimental findings is provided in Sect. 3, while Sect. 4 concludes the paper.

2 Research Methodology

A broad research of understanding and visualizing the knowledge of a model already exists in many domains. Depending of the context, different part of the model is being analysed, starting from visualizing the output of intermediate layers in generative adversarial networks [4] to interpreting convolutional and recurrent neural networks for natural language processing [2]. Of primary importance in this study is to explore the attention mechanisms [3] in the domain of sentiment analysis (sentiment classification and star detection). According to [9] attention weights should correlate with feature importance measures. In sentiment analysis, as previously stated, lexical information is proven to provide reliable features. We hypothesize that such information correlates with learned attention weights.

Following the techniques provided by [9], we have conducted several exploratory studies to examine whether attention weights correlate to word valences provided by lexical resources. For that purpose, we utilize a dataset primed for sentiment analysis and various lexicons providing word valence. We develop several deep learning models based on an attention mechanism. In the end, we analyze the weights of the trained models. Our key research questions are:

1. Does the attention layer learn the importance of information encoded in words such as valence?, and
2. To what extent do word valences correlate with attention weights?

2.1 Dataset

In this study, we utilize the Yelp[1]dataset. This dataset consists of more than 6M reviews obtained from Yelp. We apply several techniques for filtering and preprocessing the dataset. The following subsections describe in detail these techniques.

[1] https://www.yelp.com/dataset/challenge, last accesed: 13.06.2019.

Dataset Filtering. The dataset is composed of textual reviews each annotated with the ID of the user that wrote it, the business id it is written for, star rating, date and number of received votes. The total number of distinct users is approximately 1.5M. Figure 1 plots the distribution of reviews per user on logarithmic scale. The number of reviews per user fluctuates, starting with users that wrote 1 review up to users that wrote 4,129 reviews. Assuming that some users write too many reviews that are often fake, we eliminate reviews from users with a high frequency of written reviews such that only reviews written by users with review frequency in ranges [50, 500] and [10, 500] remain. For clarity, these subsets are denoted as Subset A and Subset B, respectively.

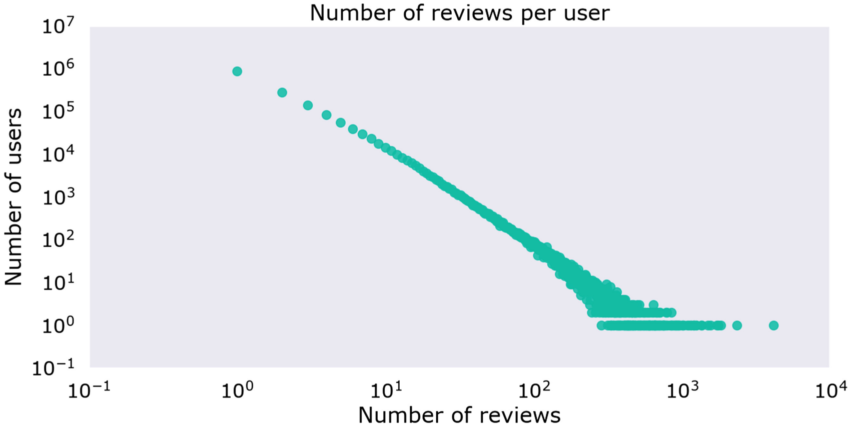

Fig. 1. Number of reviews per user.

Furthermore, we apply filtering by review length. The maximum length is 1,870, while the minimum is 1. The distribution of review length is plotted on logarithmic scale and is shown in Fig. 2. We keep only reviews with length in the range from the 25^{th} percentile to 75^{th} percentile, that is [50, 162]. The total number of reviews after filtering is summarized in Table 1.

Table 1. Number of reviews remaining after filtering.

Stars	All	Subset A (50–500)	Subset B (10–500)
1	1,002,159	39,984	165,334
2	542,394	52,255	140,111
3	739,280	116,529	238,894
4	1,468,985	224,652	475,143
5	2,933,082	216,340	678,278

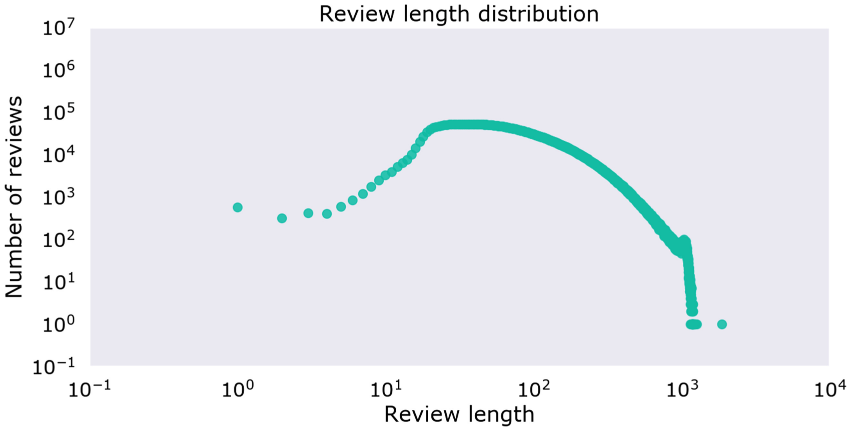

Fig. 2. Review length distribution.

Data Pre-processing. Reviews consist of multiple sentences. Since our analysis is based on words and not sentences, we consider each review as one sentence. The first step is applying necessary pre-processing steps: lower-casing and tokenizing each review. Subsequently, we extract the base form of words by applying part-of-speech tagging and lemmatization. The final pre-processing step is filtering the vocabulary. We filter the vocabulary in the following way:

- Remove words occurring in less than 15 reviews
- Remove punctuation
- Remove stopwords

The filtering is done by merely deleting tokens that satisfy the filtering criteria leading to a reduction of the number of words in a review. For Subset A, the average review length drops from 100 to 46, and from 96 to 44 for Subset B.

Sentiment Lexicons. Our analysis relies on several lexical resources i.e. lexicons.

- AFINN[2] - a list of 2,477 English words and phrases annotated with their valence rating, an integer value between -5 and 5 denoting the strength of the emotion expressed with a word.
- NRC Valence, Arousal, Dominance Lexicon [14] - a list of 20,000 English terms associated with valence, dominance, and arousal score. The score is between 0 and 1. From this lexicon, we exploit only the valence scores.
- NRC Hashtag Sentiment Lexicon [12] - a list of English n-grams annotated with real-valued score expressing the sentiment. The range is from $-\infty$ (most negative) to ∞ (most positive). The lexicon is extracted from tweets with sentiment word hashtags.

[2] http://corpustext.com/reference/sentiment_afinn.html, last accesed: 18.08.2019.

- Yelp Restaurant Sentiment Lexicon [12] - a list of English n-grams associated with real-valued score expressing the sentiment. The range is from $-\infty$ (most negative) to ∞ (most positive). The lexicon is extracted from the Yelp dataset.

We normalize scores in range [0, 1] from all lexicons except NRC Valence, Dominance, Arousal Lexicon since its scores are already in the desired range. For evaluating our proposed models, we utilize two lexicons. The first is the Yelp lexicon used as a standalone lexicon (denoted as Yelp), while the second is a combination of the other three lexicons obtained by concatenation (denoted as NRC). Each token in each review is associated with its score from both lexicons.

2.2 Deep Architectures

This section explains in detail the architectures of our proposed models. The overall architecture for all models is shown in Fig. 3. The input is a sequence of tokens composing each review. The first layer is an Embedding layer for constructing matrix representation of the input sequence. Its dimension is $m \times n$ where m is the number of tokens in the input sequence and n is the word embedding size. For the number of tokens, we use the median of the review length in the corresponding data subset, namely 45 for Subset A and 42 for Subset B.

Word embeddings represent each word in the vocabulary as a dense real-valued vector, while preserving word meanings, their relationships, and the semantic information. Pre-trained word embedding vectors already exist. We use GloVe (Global Vectors for Word Representation) [15] embedding vectors. In this study, we utilize two different word embeddings as initializers for the Embedding layer: 300-dimensional vectors pre-trained on Wikipedia and 200-dimensional vectors pre-trained on Twitter.

The matrix representation created with the Embedding layer is fed into a recurrent layer. As a recurrent layer, for all models, we use Long Short-Term Memory networks (LSTMs) with 1,024 units. The next layer is an attention mechanism inspired by [6,13]. It performs attention over hidden states of the recurrent layer. The attention outputs feature vector of attention weights for each input token which are multiplied with token representation obtained from the LSTM.

The final part of the models is Multilayer Perceptron encompassing four fully connected layers, each followed by a Dropout layer with rate 0.5. The first two layers are composed of 1,024 units, while the third and the fourth are composed of 512 and 256 units, respectively. The output is a class for the review. We train models with two different versions for the class. The first version is classifying the review as either positive or negative[3] (the model is denoted as SentDetect). The second version is a model denoted as StarDetect, which determines the number of stars that the review receives.

[3] A review is considered positive if it received at least 4 stars, and negative otherwise.

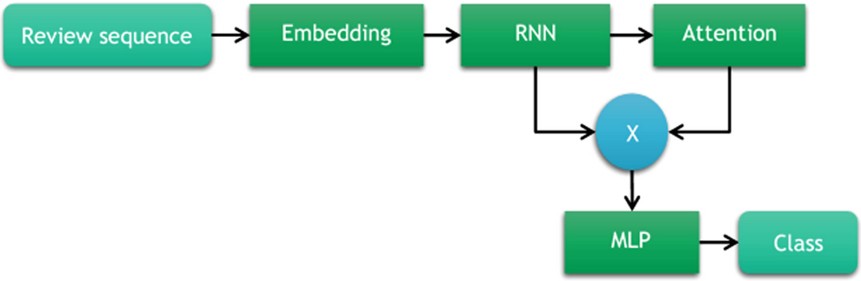

Fig. 3. Overall architecture for the models.

All models were trained with categorical cross-entropy loss function, and Adam optimizer [10] with 0.0001 learning rate and batch size 256. The training was performed on NVIDIA GeForce GTX TITAN X in 50 epochs.

3 Experiments and Discussion of Results

To test our hypothesis, we run a battery of experiments. Each subset is partitioned into training, validation and test sets in a ratio of 70:15:15. This section presents evaluation results of our models. We report on two evaluation metrics: correlation coefficient (Pearson and Kendall), and Jaccard similarity. Evaluation results presented in the following subsections are obtained on test sets.

3.1 Correlation Between Attention Weights and Word Valences

We measure the extent to which word valences correlate with attention weights by calculating correlation coefficient. The output of the attention mechanism assembles an attention matrix composed of attention weights for each word at each timestep. Single attention weight for a particular word is obtained by summing all weights for that word, creating one-dimensional attention vector for each review. With word valences we create one-dimensional valence vector with values from the NRC lexicon and one-dimensional valence vector with values from the Yelp lexicon. All three vectors have the same length that is the minimum of the number of words in the review and the padding size. In fact, we do not take into account words that have not contributed to the final prediction when the length is greater than or equal to the padding size, and omit padded zeros when the length is less than the padding size.

Table 2 summarizes our findings about SentDetect and StarDetect model, respectively. We report on two correlation metrics - Pearson correlation coefficient and Kendall's tau coefficient. Correlation coefficients are calculated between attention vector and: (1) NRC valence vector and (2) Yelp valence vector for each review in the set. The final coefficient is computed by averaging over the obtained values. The observed correlations are near 0[4] indicating no correspondence between word valences and attention weights.

Table 2. Correlation coefficient for SentDetect and StarDetect models. Pearson correlation coefficient - r, Kendall correlation coefficient - τ. GloVe embeddings pre-trained on Wikipedia - w, GloVe embeddings pre-trained on Twitter - t.

			NRC		Yelp	
			r	τ	r	τ
SentDetect	Subset A	w	−0.00646	−0.00482	−0.00026	−0.00201
		t	0.0246	0.02369	0.02872	0.02428
	Subset B	w	−0.01277	−0.00633	−0.00368	−0.00185
		t	−0.00899	−0.00957	−0.00072	−0.00193
StarDetect	Subset A	w	−0.02745	−0.01291	−0.03327	−0.01976
		t	0.00124	−0.00632	0.04243	0.03106
	Subset B	w	−0.00043	0.00332	0.00473	0.00261
		t	−0.00078	0.00123	0.00203	0.0022

3.2 Jaccard Similarity

We further investigate whether words with high attention weights are those having high lexical valences. For this purpose, words are sorted by two criteria: attention weight and valence. The first criteria simply assumes that higher weight value implies high attention weight. For the second criteria we need to consider two different aspects, whether the review is positive or negative. We assume that higher lexical value implies higher valence if the review is positive or has more than three stars, while lower lexical value implies higher valence if the review is negative or has less than four stars.

After that, we compute the overlap of the set of words with highest attention weights and the set of words with highest valences. The overlap is defined as Jaccard similarity between the two sets of words. We have no clear definition of what highest weights and highest valences stand for i.e. how many words are those with highest weights and how many words are those with highest valences. Therefore, we report Jaccard similarity at a specific cutoff. For instance, if the cutoff is set to 5, Jaccard similarity between the set of first 5 words with highest attention weights and the set of first 5 words with highest lexical valences is computed. Note here that computing Jaccard similarity at cutoff equal to padding size (45 for Subset A and 42 for Subset B) leads to value 1 for similarity since, although having different ordering, both sets of words are equal.

The Jaccard similarity is calculated at four different cutoff sizes: 5, 10, 15 and 20. Table 3 reports summary statistics for evaluation on the NRC lexicon,

[4] Value 0 implies that the two sets are not correlated.

while Table 4 reports summary statistics for evaluation on the Yelp lexicon. For the first cutoff size, 5, the similarity is approximately 0.08 which indicates that the average number of mutual words is less than 1. Values increase as the cutoff size increases which is reasonable since the sets are expanding too. We could hypothesize that both sets will have more mutual words by taking into account more words. For example, the similarity at cutoff 20 is approximately 0.4 indicating that the average number of mutual words is around 8. According to the findings, we can conclude that word valence, despite playing significant role in sentiment analysis, is not the most significant information in attention-based deep learning models.

Table 3. Jaccard similarity for SentDetect and StarDetect models evaluated on the NRC lexicon. GloVe embeddings pre-trained on Wikipedia - *w*, GloVe embeddings pre-trained on Twitter - *t*.

			NRC			
			First 5	First 10	First 15	First 20
SentDetect	Subset A	w	0.0803	0.1678	0.27335	0.40283
		t	0.0802	0.1676	0.27421	0.4039
	Subset B	w	0.08722	0.17823	0.28883	0.42577
		t	0.0891	0.17649	0.28783	0.42621
StarDetect	Subset A	w	0.07991	0.16722	0.27439	0.40549
		t	0.08593	0.17326	0.27772	0.40707
	Subset B	w	0.08604	0.17669	0.28695	0.42336
		t	0.08533	0.17594	0.28636	0.42271

Table 4. Jaccard similarity for SentDetect and SentDetect models evaluated on the Yelp lexicon. GloVe embeddings pre-trained on Wikipedia - *w*, GloVe embeddings pre-trained on Twitter - *t*.

			Yelp			
			First 5	First 10	First 15	First 20
SentDetect	Subset A	w	0.08295	0.1686	0.27308	0.4021
		t	0.083	0.16938	0.27508	0.40476
	Subset B	w	0.08932	0.179	0.29013	0.42665
		t	0.08855	0.17744	0.28812	0.42561
StarDetect	Subset A	w	0.08249	0.16777	0.27374	0.40507
		t	0.09034	0.17525	0.27799	0.40675
	Subset B	w	0.08715	0.17774	0.28723	0.42331
		t	0.08719	0.17702	0.28676	0.42342

4 Conclusion

In this paper, we aim to explore the attention mechanishm and investigate its learned weights. We proposed a deep learning architecture based on attention mechanism and trained different models on two sentiment analysis tasks: review sentiment classification and review star detection.

Information about affective words and phrases, and their assigned valence scores provide reliable features for machine learning models. Moreover, when incorporated into deep learning models, such lexical information improves the performance of the models. The key task in this paper is to examine whether the attention mechanism is able to capture the importance of word valences and whether the attention weights correlate to word valences provided by lexical resources.

By performing various experiments, the findings suggest that word valence is not the most significant information in attention-based deep learning models. Obtained evaluation results lead to conclusion that word valences do play significant role in sentiment analysis, but possibly models rely upon other dimensions perhaps not distinguishable by humans.

References

1. Abdul-Mageed, M., Ungar, L.: Emonet: fine-grained emotion detection with gated recurrent neural networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (vol. 1: Long Papers), pp. 718–728 (2017)
2. Alishahi, A., Chrupala, G., Linzen, T.: Analyzing and interpreting neural networks for NLP: a report on the first blackboxnlp workshop. Nat. Lang. Eng. (2019)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2014)
4. Bau, D., et al.: GAN dissection: visualizing and understanding generative adversarial networks (2018)
5. Chen, H., Sun, M., Tu, C., Lin, Y., Liu, Z.: Neural sentiment classification with user and product attention. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1650–1659 (2016)
6. Giannakopoulos, A., Antognini, D., Musat, C., Hossmann, A., Baeriswyl, M.: Dataset construction via attention for aspect term extraction with distant supervision. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 373–380. IEEE (2017)
7. Gievska, S., Koroveshovski, K., Chavdarova, T.: A hybrid approach for emotion detection in support of affective interaction. In: 2014 IEEE International Conference on Data Mining Workshop, pp. 352–359. IEEE (2014)
8. Hasan, M., Rundensteiner, E., Agu, E.: Emotex: detecting emotions in Twitter messages. In: Proceedings of the Sixth ASE International Conference on Social Computing (SocialCom 2014). Academy of Science and Engineering (ASE) (2014)
9. Jain, S., C. Wallace, B.: Attention is not explanation. CoRR (2019)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014)

11. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: NRC-Canada-2014: detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 437–442 (2014)
12. Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. J. Artif. Intell. Res. **50**, 723–762 (2014)
13. Lin, Z., et al.: A structured self-attentive sentence embedding. CoRR (2017)
14. Mohammad, S.: Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (vol. 1: Long Papers), pp. 174–184 (2018)
15. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
16. Shin, B., Lee, T., Choi, J.D.: Lexicon integrated CNN models with attention for sentiment analysis. In: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 149–158 (2017)
17. Stojanovska, F., Toshevska, M., Gievska, S.: Explorations into deep neural models for emotion recognition. In: Kalajdziski, S., Ackovska, N. (eds.) ICT 2018. CCIS, vol. 940, pp. 217–232. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00825-3_19
18. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1422–1432 (2015)