Attention-Based Sentiment Analysis on Movie Reviews

Li-Hen Chen (UIN:928003907), Yun He (UIN:326005850), Qifan Li (UIN:127004551)
Fan Yang (UIN:525004621), Yining Zhou (UIN:927009507)

***Abstract*—In this project, We classify the reviews from movie websites into positive and negative comments using dierent machine learning models as well as deep learning models, including logistic regression, linear/non-linear SVM, Random Forest, CNN , bidirectional GRU, etc. We get the dataset from SST-2 and we use pre-trained word-embedding tools Glove for obtaining embedding vector representations for words. Besides, we also provide interpretations to the sentiment classication task with attention-based method. The results turn out that Bi-RNN gives the best accuracy performance.**

***Index Terms*—Text classification, sentiment analysis, attentional neural networks.**

I. Introduction

In the web era, sentiment classification has been becoming a significant problem nowadays, due to the tons of online available textual information. To better facilitate human in doing online textual analysis for data mining, machine learning and deep learning techniques have been widely utilized in real-world applications. In this project, we classify the reviews from movie websites into positive and negative comments. We used different machine learning models as well as deep learning models, including Logistic Regression, linear/nonlinear SVM, Random Forest, CNN, bidirectional GRU, etc. We get the dataset from SST-2 and we use pre-trained word-embedding tools Glove for obtaining embedding vector representations for words. However, most of those applied learning systems typically lacks interpretability, which largely hinders developers to know whether their deployed models are reasonable or not. Therefore, we also provide interpretations to the sentiment classification task with attention-based method. The results turn out that Bi-RNN gives the best accuracy performance.

II. Literature Review

Here are many types and flavors of sentiment analysis and SA tools range from systems that focus on polarity (positive, negative, neutral) to systems that detect feelings and emotions (angry, happy, sad, etc) or identify intentions (e.g. interested v. not interested). In the following section, well cover the most important ones.

A. Fine-grained Sentiment Analysis [1-3]

Sometimes you may be also interested in being more precise about the level of polarity of the opinion, so instead of just talking about positive, neutral, or negative opinions you could consider the following categories: *Very positive, Positive, Neutral, Negative, Very negative*. This is usually referred to as fine-grained sentiment analysis. This could be, for example, mapped onto a 5-star rating in a review, e.g., Very Positive = 5 stars and Very Negative = 1 star. Some systems also provide different flavors of polarity by identifying if the positive or negative sentiment is associated with a particular feeling, such as, anger, sadness, or worries (i.e. negative feelings) or happiness, love, or enthusiasm (i.e. positive feelings).

B. Emotion Detection [4-6]

Emotion detection aims at detecting emotions like, happiness, frustration, anger, sadness, and the like. Many emotion detection systems resort to lexicons (i.e. lists of words and the emotions they convey) or complex machine learning algorithms. One of the downsides of resorting to lexicons is that the way people express their emotions varies a lot and so do the lexical items they use. Some words that would typically express anger like shit or kill (e.g. in your product is a piece of shit or your customer support is killing me) might also express happiness (e.g. in texts like This is the shit or You are killing it).

C. Aspect-based Sentiment Analysis [7-9]

Usually, when analyzing the sentiment in subjects, for example products, you might be interested in not only whether people are talking with a positive, neutral, or negative polarity about the product, but also which particular aspects or features of the product people talk about. That's what aspect-based sentiment analysis is about. In our previous example: "The battery life of this camera is too short." The sentence is expressing a negative opinion about the camera, but more precisely, about the battery life, which is a particular feature of the camera.

D. Intent Analysis [10-12]

Intent analysis basically detects what people want to do with a text rather than what people say with that text. Look at the following examples: Your customer support is a disaster. Ive been on hold for 20 minutes, I would like to know how to replace the cartridge, Can you help me fill out this form?. A human being has no problems detecting the complaint in the first text, the question in the second text, and the request in the third text. However, machines can have some problems to identify those. Sometimes, the intended action can be inferred from the text, but sometimes, inferring it requires some contextual knowledge.

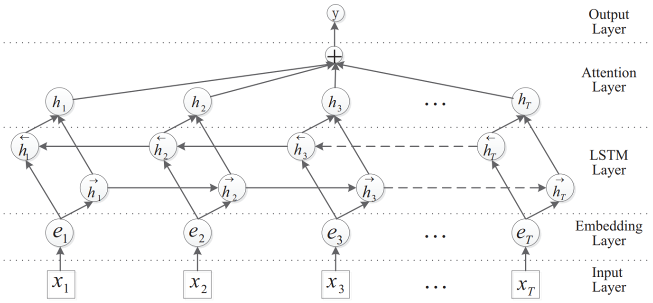


Fig. 1. The overall architecture of the implemented attentional model [16].

E. Multilingual Sentiment Analysis [13-15]

Multilingual sentiment analysis can be a difficult task. Usually, a lot of preprocessing is needed and that preprocessing makes use of a number of resources. Most of these resources are available online (e.g. sentiment lexicons), but many others have to be created (e.g. translated corpora or noise detection algorithms). The use of the resources available requires a lot of coding experience and can take long to implement. An alternative to that would be detecting language in texts automatically, then train a custom model for the language of your choice (if texts are not written in English), and finally, perform the analysis.

III. Problem Formulation

To better facilitate human in doing online textual analysis for data mining, machine learning and deep learning techniques have been utilized in many real-world applications. However, most of those applied learning systems typically lacks interpretability, which largely hinders developers to know whether their deployed models are reasonable or not. In this project, we aim to provide some interpretation to the sentiment classication task beyond the prediction results, so as to help developers better understand the system they deploy.

IV. Proposed Solution

In this paper, we use the bidirectional RNN to learn the characteristics of the text, because the meaning of a word is not only related to the text content in front of it, but also related to the text content behind it. We use the bidirectional RNN method to implement the text represented from the learning, and then the two directions to learn the feature vector spliced together, this as a text vector, so that relative to the unidirectional RNN, the eigenvector of the semantics is more comprehensive and rich. At the same time, we add a mechanism of attention to the network model, for each word to learn a weight, making the key words have a heavier weight, and non-key words have a lighter weight, which can make important features become more prominent. Fig. 1 shows the overall architecture of the model.

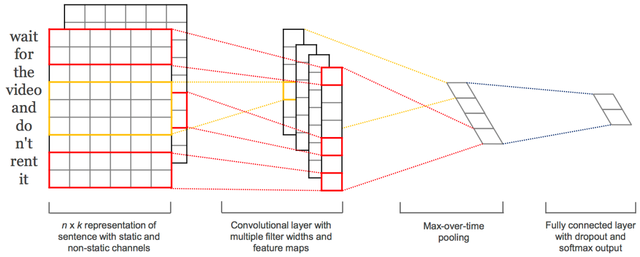

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Fig. 2. The overall architecture of the implemented CNN model [17].

Attentive neural networks have recently demonstrated success in a wide range of tasks ranging from question answering, machine translations, speech recognition, to image captioning. We employ the attention mechanism for text classification tasks. Let **H** be a matrix consisting of output vectors $[h_1, h_2, \cdots, h_T]$ that the LSTM layer produced, where T is the sentence length. The representation r of the sentence is formed by a weighted sum of these output vectors:

$$M = \tanh(H), \tag{1}$$

$$\alpha = \mathrm{softmax}(w^T M), \tag{2}$$

$$r = H\alpha^T, \tag{3}$$

where $H \in R^{d^w \times T}$, d^w is the dimension of the word vectors, w is a trained parameter vector and w^T is a transpose. The dimension of w, α, r is d^w, T, d^w separately. We obtain the final sentence-pair representation used for classification from:

$$h^* = \tanh(r). \tag{4}$$

For the CNN model we employed, the first layers embeds words into low-dimensional vectors. The next layer performs convolutions over the embedded word vectors using multiple filter sizes. For example, sliding over 3, 4 or 5 words at a time. Next, we max-pool the result of the convolutional layer into a long feature vector, add dropout regularization, and classify the result using a softmax layer. A multi-channel convolutional neural network for document classification involves using multiple versions of the standard model with different sized kernels. This allows the document to be processed at different resolutions or different n-grams (groups of words) at a time, whilst the model learns how to best integrate these interpretations. The overall structure is shown in Fig. 2.

Here we depict three filter region sizes: 2, 3 and 4, each of which has 2 filters. Every filter performs convolution on the sentence matrix and generates (variable-length) feature maps. Then 1-max pooling is performed over each map, i.e., the largest number from each feature map is recorded. Thus a univariate feature vector is generated from all six maps, and these 6 features are concatenated to form a feature vector for the penultimate layer. The final softmax layer then receives this feature vector as input and uses it to classify the sentence; here we assume binary classification and hence depict two possible output states.

Basically, the above structure is implementing what we have done above with bigram filters, but not only to bigrams but also to trigrams and fourgrams. However, this is not linearly stacked layers, but parallel layers. And after convolutional layer and max pooling layer, it simply concatenated max pooled result from each of bigram, trigram, and fourgram, then build one output layer on top of them.

V. Data Description

We use the dataset from SST-2 which consists of sentences extracted from movie reviews and human annotations of their sentiment. Given a sentence, the task is to determine the sentiment of the sentence (positive or negative). In our project, we use pre-trained word-embedding tools Glove for obtaining embedding vector representations for words, use the ”train.tsv” file for training and the ”dev.tsv” file for validating the performance of our methods.

VI. Results

In the Fig. 3 and Fig. 4 that most of the words are below 10,000 on both X-axis and Y-axis, and we cannot see meaningful relations between negative and positive frequency. However, if we combine harmonic mean of rate CDF and frequency CDF to interpret sentiment data, it has created an interesting pattern on the plot. If a data point is near to the upper left corner, it is more positive, and if it is closer to the bottom right corner, it is more negative. With the figure, we can see what token each data point represents by hovering over the points. Not every point has the correlation as we expect, we believe thats the reason why our accuracy is around 85% eventually.

The pre-trained model we used is Glove300d, which is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. The results are shown in the below table. The models we train include some deep learning models such as bidirectional GRU + attention, bidirectional RNN, CNN, and NN and some baseline models such as SVM, Random Forest and Logistic Regression. The classification performance is shown in the following TABLE I.

TABLE I
Classification Performance

Model	Accuracy
Bi-GRU+Attention	86.35%
Bi-RNN	86.04%
CNN	84.65%
MLP	80.53%
Linear SVM	67.43%
Non-linear SVM	69.15%
Random Forest	66.83%
Logistic Regression	68.23%

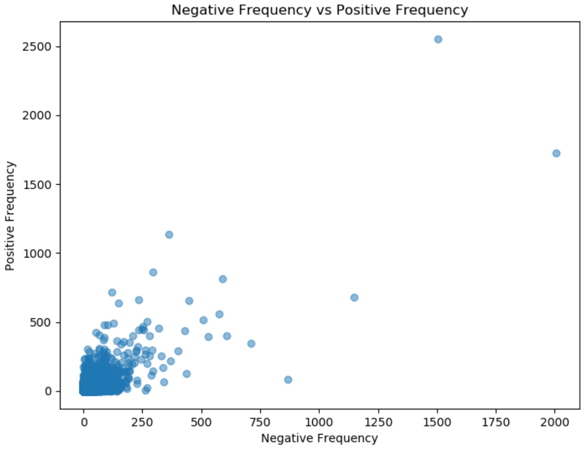


Fig. 3. Correlation of words frequency.

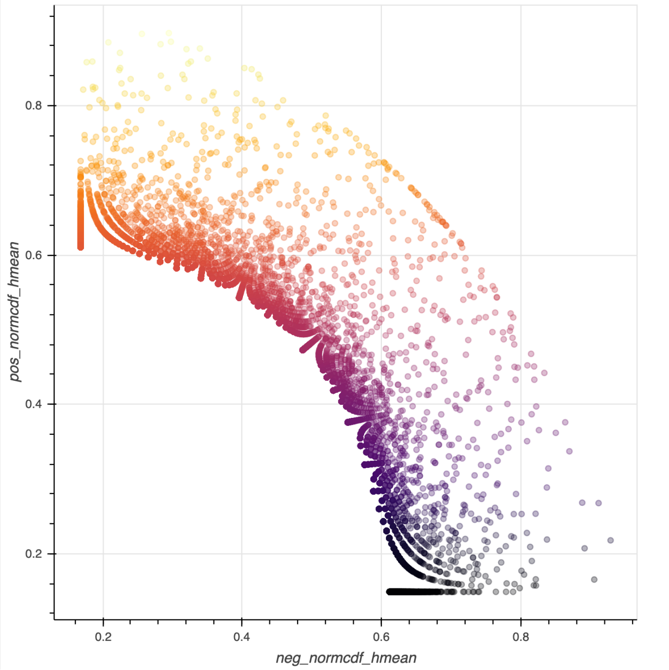
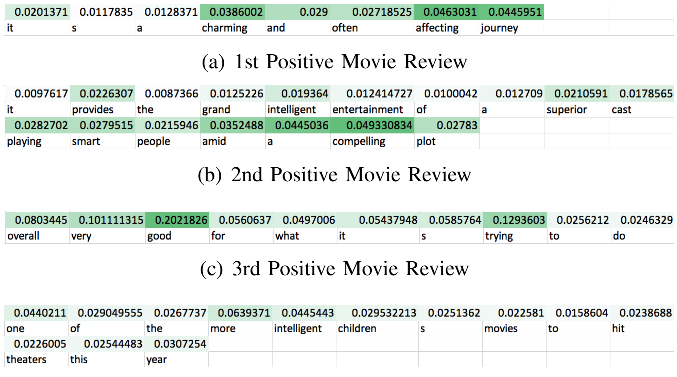


Fig. 4. Frequency CDF Harmoor nic mean.

We also give several case studies as shown. As discussed before, our model can not only make a determination of sentiment polarity for a review, but also assign attention scores to the tokens in that review. The attention scores can be interpreted as the weights or importances of the tokens in terms of their informative to the sentiment polarity of the review. In this section, we will discuss the performance of our model on several cases (i.e., movie reviews) and see if our model can provide reasonable interpretation for the result. We select eight typical reviews that our model gives a correct label, where the word tokens and their corresponding attention scores are shown below.

The heatmap of the attention scores on positive reviews is shown in Figure 5 while the heatmap of the attention scores on negative reviews is shown in Figure 6. The darker the color is, the higher the attention score and the more important of the token is. It should be noted that we have a bidirectional GRU layer before the attention layer, which means the input for the attention layer is contextual word embedding and thus the



0.0201371	0.0117835	0.0128371	0.0386002	0.029	0.02718525	0.0463031	0.0445951		
it	s	a	charming	and	often	affecting	journey		

(a) 1st Positive Movie Review

0.0097617	0.0226307	0.0087366	0.0125226	0.019364	0.012414727	0.0100042	0.012709	0.0210591	0.0178565
it	provides	the	grand	intelligent	entertainment	of	a	superior	cast
0.0282702	0.0279515	0.0215946	0.0352488	0.0445036	0.049330834	0.02783			
playing	smart	people	amid	a	compelling	plot			

(b) 2nd Positive Movie Review

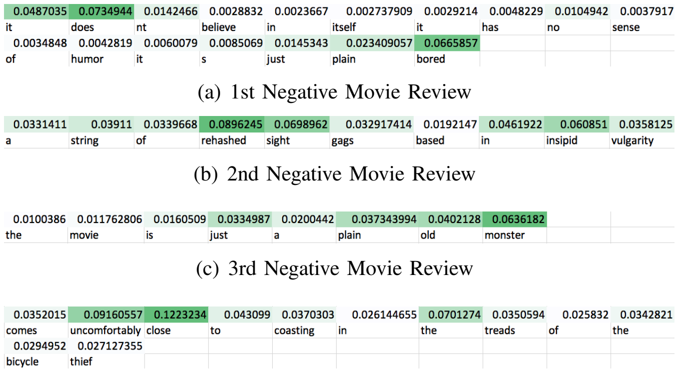
0.0803445	0.101111315	0.2021826	0.0560637	0.0497006	0.05437948	0.0585764	0.1293603	0.0256212	0.0246329
overall	very	good	for	what	it	s	trying	to	do

(c) 3rd Positive Movie Review

0.0440211	0.029049555	0.0267737	0.0639371	0.0445443	0.029532213	0.0251362	0.022581	0.0158604	0.0238688
one	of	the	more	intelligent	children	s	movies	to	hit
0.0226005	0.02544483	0.0307254							
theaters	this	year							

(d) 4th Positive Movie Review

Fig. 5. Case study of positive reviews.



0.0487035	0.0734944	0.0142466	0.0028832	0.0023667	0.002737909	0.0029214	0.0048229	0.0104942	0.0037917
it	does	nt	believe	in	itself	it	has	no	sense
0.0034848	0.0042819	0.0060079	0.0085069	0.0145343	0.023409057	0.0665857			
of	humor	it	s	just	plain	bored			

(a) 1st Negative Movie Review

0.0331411	0.03911	0.0339668	0.0896245	0.0698962	0.032917414	0.0192147	0.0461922	0.060851	0.0358125
a	string	of	rehashed	sight	gags	based	in	insipid	vulgarity

(b) 2nd Negative Movie Review

0.0100386	0.011762806	0.0160509	0.0334987	0.0200442	0.037343994	0.0402128	0.0636182		
the	movie	is	just	a	plain	old	monster		

(c) 3rd Negative Movie Review

0.0352015	0.09160557	0.1223234	0.043099	0.0370303	0.026144655	0.0701274	0.0350594	0.025832	0.0342821
comes	uncomfortably	close	to	coasting	in	the	treads	of	the
0.0294952	0.027127355								
bicycle	thief								

(d) 4th Negative Movie Review

Fig. 6. Case study of negative reviews.

attention scores are also contextual. In other words, the same word will have different attention scores in different context (i.e., reviews).

We observe that our model is able to assign high weights to key words in terms if their information to determine the sentiment polarity. For example, in Figure 5(a), the review is positive where “charming” and “affecting” obtain relatively high attention scores: 0.038 and 0.046 respectively while some stopwords like “a” and “and” just get weights as 0.012 and 0.029. The same observation is also obtained in negative reviews. For example, in Figure 6(d), “bored” get a high attention score 0.066. Therefore, the attention scores can reflect the importance of word tokens and can be used to provide reasons why our model make such a determination. In conclusion, our model can explainably classify the sentiment polarity of movie reviews.

VII. Conclusions

In this project, we implement an interpretable deep model for conventional sentiment classification task, using the attention-based bidirectional RNN architecture. With the aid of the attention weights corresponding to each word, end-users could know how the trained model makes the classification beyond the prediction results. By conducting a series of experiments, we observe that our interpretable attention-based model could achieve a competitive performance among all other baselines, including other non-interpretable deep models. Meanwhile, our model could generate explanations for each new coming instance (movie review), so as to help users understand of contribution of each word for sentiment classification. Besides, with some case studies, we found that our generated explanations is sensible, which effectively captures the important words for sentiment analysis.

Acknowledgment

At the end of this report, we would like to give a great thanks to Dr. Chaspari for the great lecture given during this whole semester. We definitely learned a lot with those wonderful lectures and useful exams. During this project, we also have a great opportunity to use the knowledge we learned, and implemented some of the new models that have been proposed recently. This course is certainly a good introduction for the fundamentals of machine learning, and it will facilitate us to go further within this research area. Thank you!

References

[1] Kiritchenko, Svetlana, and Saif M. Mohammad. ”Capturing reliable fine-grained sentiment associations by crowdsourcing and best-worst scaling.” arXiv preprint arXiv:1712.01741 (2017).

[2] Zhou, Feng, et al. ”Fine-grained facial expression analysis using dimensional emotion model.” arXiv preprint arXiv:1805.01024 (2018).

[3] Guzman, Emitza, and Walid Maalej. ”How do users like this feature? a fine grained sentiment analysis of app reviews.” 2014 IEEE 22nd international requirements engineering conference (RE). IEEE, 2014.

[4] Soleymani, Mohammad, et al. ”Analysis of EEG signals and facial expressions for continuous emotion detection.” IEEE Transactions on Affective Computing 7.1 (2016): 17-28.

[5] Jerauld, Robert. ”Wearable emotion detection and feedback system.” U.S. Patent No. 9,019,174. 28 Apr. 2015.

[6] Fernndez-Caballero, Antonio, et al. ”Smart environment architecture for emotion detection and regulation.” Journal of biomedical informatics 64 (2016): 55-73.

[7] Ruder, Sebastian, Parsa Ghaffari, and John G. Breslin. ”A hierarchical model of reviews for aspect-based sentiment analysis.” arXiv preprint arXiv:1609.02745 (2016).

[8] Poria, Soujanya, et al. ”Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis.” 2016 international joint conference on neural networks (IJCNN). IEEE, 2016.

[9] Ma, Yukun, Haiyun Peng, and Erik Cambria. ”Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM.” Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[10] Li, Shijin, Minchen Zhu, and Yanbin Qiu. ”Attack Intent Analysis Method Based on Attack Path Graph.” Proceedings of the 8th International Conference on Communication and Network Security. ACM, 2018.

[11] Li, Shijin, Minchen Zhu, and Yanbin Qiu. ”Attack Intent Analysis Method Based on Attack Path Graph.” Proceedings of the 8th International Conference on Communication and Network Security. ACM, 2018.

[12] Newman, Spencer H. ”Unreasonably Risky: Why a Negligence Standard Should Replace the Bankruptcy Code’s Fraudulent Intent Analysis for Gambling Debts.” UNLV Gaming Law Journal 8.2 (2018): 7.

[13] Dashtipour, Kia, et al. ”Multilingual sentiment analysis: state of the art and independent comparison of techniques.” Cognitive computation 8.4 (2016): 757-771.

[14] Proksch, SvenOliver, et al. ”Multilingual sentiment analysis: A new approach to measuring conflict in legislative speeches.” Legislative Studies Quarterly 44.1 (2019): 97-131.

[15] Araujo, Matheus, et al. ”An evaluation of machine translation for multilingual sentence-level sentiment analysis.” Proceedings of the 31st Annual ACM Symposium on Applied Computing. ACM, 2016.

[16] Zhou, Peng, et al. ”Attention-based bidirectional long short-term memory networks for relation classification.” Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vol. 2. 2016.

[17] Kim, Yoon. ”Convolutional neural networks for sentence classification.” arXiv preprint arXiv:1408.5882 (2014).