INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Explaining Sentiment Classification

Marvin Rajwadi[1,2], *Cornelius Glackin*[2], *Julie Wall*[1], *Gérard Chollet*[2], *Nigel Cannings*[2]

[1]School of Architecture, Computing and Engineering, University of East London, U.K.
[2]Intelligent Voice Ltd., London, U.K.

marvin.rajwadi@intelligentvoice.com, neil.glackin@intelligentvoice.com, j.wall@uel.ac.uk, gerard.chollet@intelligentvoice.com, nigel.cannings@intelligentvoice.com

Abstract

This paper presents a novel 1-D sentiment classifier trained on the benchmark IMDB dataset. The classifier is a 1-D convolutional neural network with repeated convolution and max pooling layers. The main contribution of this work is the demonstration of a deconvolution technique for 1-D convolutional neural networks that is agnostic to specific architecture types. This deconvolution technique enables text classification to be explained, a feature that is important for NLP-based decision support systems, as well as being an invaluable diagnostic tool.

Index Terms: Explainability, Interpretability, Sentiment Classification, 1-D Convolutional Neural Networks

1. Introduction

The recent GDPR regulations [1] have important implications for the deployment of real-world customer-facing AI systems. Under these regulations, humans have a right to have decisions explained to them, and this has serious implications for the suitability and use of automated AI systems. Coping strategies based on counter-factual explanations [2] have been posited on the one hand, and on the other there is a perception that GDPR is simply incompatible [3] with many of the current practices of deep learning and AI, certainly in their current state. If one considers the rise of end-to-end speech recognition systems as an example, it is commonly understood that such systems have simplified the process of training Automatic Speech Recognition systems. However, if we legally have to explain how such a system arrived at a particular transcription, this would be currently difficult, if not impossible to do, as we would need to untangle the decision process from the black-box architecture. As such autonomous deep learning based systems replace humans in the decision making process, it now becomes necessary for AI-based autonomous systems to explain themselves [4].

Arguably by accident, some deep learning architectures have transparency with regard to how they arrive at their classifications, for example the attention mechanism [5] that maps encoder and decoder states provides such insight in neural machine translation. Recent advances with attention have seen a move away from recurrent units entirely for sequence-to-sequence architectures formed entirely of attention mechanisms [6]. The attention mechanism effectively turns the sequence problem into a spatial representation, enabling long-range dependencies in sequences to be related more effectively. Similarly, activation patterns in convolutional neural network (CNN) architectures can provide insight into CNN classification. Such approaches, termed deconvolution [7], effectively enable the projection of features back to the input space, providing insight into what the network *sees*. For image classification for self-driving cars using CNNs [8] and for hybrid CNN-RNN approaches for image captioning [9], this approach provides significant diagnostic information. Interestingly, for the image captioning implementation, the explainability is provided by the combined efforts of the attention mechanism and deconvolution functionality.

There are many approaches that try to reverse engineer the inferencing of CNNs, most notable is the recent Grad-CAM implementation [10], which is based on guided backpropagation of activation maps. However, most of the activation map-based approaches require intimate knowledge of the particular CNN architecture and the approach needs to be tailored for different architectures.

The method of text deconvolution by occlusion proposed in this paper was inspired by [7], where systematically regions in the input image are occluded by a gray square. That image is inferenced with the trained model, and the shift in classification accuracy for a particular class is recorded. By overlaying the grid of classification accuracies corresponding to the pixel position of the centre of the occluded squares, one can determine the regions of the input image that contribute the most to the classification of the image as a whole. This approach is computationally demanding in that in order to understand a classification of an image one needs to perform classification of that same input image each with a different occluded region. However, the classification can be run in parallel with multiprocessing. Similarly, larger strides of the occluded region can be used to limit the computational overhead for 2-D deconvolution by occlusion.

In this paper we will be taking a similar approach to the deconvolution by occlusion approach for 2-D CNNs and applying it to the 1-D text classification problem domain.

2. Sentiment Classification

There is a trade-off between the number of out-of-vocabulary words and vocabulary size that is a significant problem in sequence-to-sequence tasks [11]. In our text classification task, we in part address this problem using word embeddings, and also by capping the number of words in the vocabulary. Whilst for machine translation this limit on vocabulary size might not be suitable, it is less of an issue in this domain as the vocabulary size for chat text in the IMDB corpus is significantly smaller than typical written text vocabularies. With this in mind, we choose a word rather than character-based representation as used in [12], and harness the embedding layer to limit this dimensionality problem. We also avoid the unnecessarily long sequences associated with character-based encoding.

The remainder of this section outlines our approach to sentiment classification. First we introduce the well-known IMDB dataset, and describe the data preparation performed before presenting the 1-D CNN architecture that we will subsequently base our deconvolutional work on.

http://dx.doi.org/10.21437/Interspeech.2019-2743

2.1. IMDB dataset

The IMDB dataset [13] also referred to as the Large Movie Dataset is a binary sentiment analysis dataset consisting of 50,000 highly polar movie reviews, labelled as good or bad reviews. The data was gathered by Stanford researchers and was split evenly between training and testing data with 25,000 examples for each set, and labeled as positive or negative. The dataset contains an even number of positive and negative reviews. A negative review has a score ≤ 4 out of 10, and a positive review has a score ≥ 7 out of 10. No more than 30 reviews are included per movie. Models are typically evaluated based on accuracy, which is sufficient since the data is balanced.

2.2. Data Preparation

Data pre-processing is the most important aspect of training a model, since the quality of the resulting model is directly correlated with the quality of the data [14]. Raw-text can contain significant noise in the form of punctuation and whitespace. Hence, the first step in pre-processing is cleaning the raw review text, replacing upper-case characters with lowercase, and removing punctuation and whitespace using regular expressions [15]. The second step in pre-processing is to convert the clean review text into an input appropriate for our defined model, using tokenization. Tokenization is the process of splitting text (strings) into a list of tokens. In this work, we used Tokenizer, which is a tool available in the Natural Language Toolkit (NLTK) [16]. After tokenizing the review text, we use the tokenizer function to create a word index dictionary. The tokenizer function assigns a unique number to the whole vocabulary used in the entire dataset in order of the most frequently used words. We have 88,585 unique words used in the IMDB dataset. The review is then converted into a list of corresponding word indices and then padded with 0's to a fixed length for training, as the input must be of similar length. We keep the input sequence to a maximum length of 2,000 words and for this particular data-set we use a vocabulary size of 4,000.

2.3. 1-D CNN Architecture

Historically, successively deeper approaches to 2-D CNN architectures is arguably the main reason for success reported by winners of the ImageNet competition [17]. This pursuit of deeper models has led to a huge surge in applications, and innovative approaches to minimizing parameters, improving training efficiency, and has led to better and more robust architectures in the image classification [18] domain. The prosperity of deep networks comes from their ability to learn hierarchical feature representations from data which varies in complexity from pixels and lines, all the way to highly complex shapes and objects. However that is not the case when dealing with word representations in the text classification domain. Here deep networks perform poorly for this particular problem, as the impact of depth in the Natural Language Processing (NLP) domain is still unclear [19].

There has been a lot of debate when it comes to to time sequence related classifications, what is better a CNN or an RNN? A recent publication in 2017 by Facebook AI demonstrated results using a fully convolutional translation model which outperforms an LSTM based model in performance and reported a speed up of 9x [20]. It is also claimed that due to their hierarchical nature, that CNN architectures learn compositional structure more easily.

In text classification, a previous state-of-the-art approach used hierarchical attention networks [21] to learn long text sequences. This approach made use of self-attention, i.e. word-by-word relationships between words in the same sentence, and exploiting that information to capture the internal structure of the sentence. Similarly, hierarchical convolutional attention networks [22] use self-attention to address the issues with the sliding window length of Kim's CNN architecture [23]. In the original implementation, Kim uses a sliding window encompassing 4 or 5 words. It is claimed that this means that Kim's approach is incapable of learning linguistic patterns beyond this 4 or 5 word window size. In this work we do not use a sliding window to maintain a consistent sequence length, neither do we rely on the use of pre-learned embeddings. Instead we employ variable length sequences encoded by an embedding layer that learns embeddings during training, and masks padded elements appropriately from the loss function. We therefore have *no* sliding window limitation on the length of the sequences that we can learn.

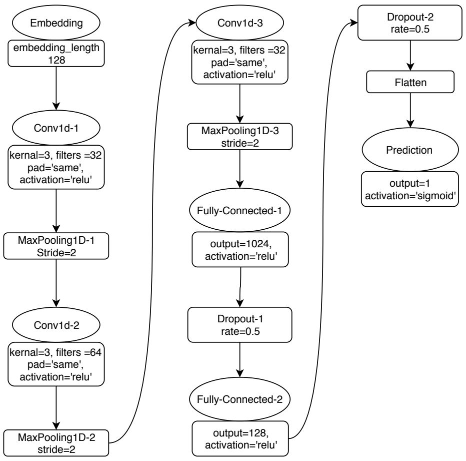

Embedding
embedding_length 128
Conv1d-1
kernal=3, filters =32 pad='same', activation='relu'
MaxPooling1D-1 Stride=2
Conv1d-2
kernal=3, filters =64 pad='same', activation='relu'
MaxPooling1D-2 stride=2
Conv1d-3
kernal=3, filters =32 pad='same', activation='relu'
MaxPooling1D-3 stride=2
Fully-Connected-1
output=1024, activation='relu'
Dropout-1 rate=0.5
Fully-Connected-2
output=128, activation='relu'
Dropout-2 rate=0.5
Flatten
Prediction
output=1 activation='sigmoid'

Figure 1: *Architecture of the 1-D CNN Sentiment Classifier*

The architecture of the 1-D text classifier in this work is shown in Figure 1. Our architecture validates the conclusion that a relatively shallow CNN with very little hyper-parameter tuning and static vectors can achieve competitive results on sentence-level classification tasks [23]. As can be seen in the figure, our model consists of an embedding layer followed by three Convolutional 1-D layers chained to a ReLU activation layer [24] and MaxPooling 1-D layers. We keep the padding the same throughout all the layers, and we have two fully-connected layers followed by a ReLU activation and a Dropout layer [25] for regularisation, which are attached to the last Maxpooling 1-D layer. The model is compiled using a binary cross-entropy loss function and an RMSprop optimizer with a fixed 1e-3 learning rate and mini batch size of 128. The dense fully connected layers have Sigmoid activation which outputs a number between 0 (negative) to 1 (positive).

Before the data goes into the first CNN layer, it passes through the Embedding layer which trains on that data and is a crucial layer when dealing with text. The Embedding layer converts each element in the word index sequence input to a simple vector representation, which in turn allows faster and more efficient processing with text data. However the padded input contains a lot of 0's and hence the embedding layers also masks those numbers from the loss function during training. The learned embeddings from the embedding layer should not be confused with the embeddings that Glove [26] or word2vec [27] learn. These related embeddings are trained to capture semantic similarity whilst the Embedding layer in this work outputs embeddings that are configured purely for classification purposes on the dataset itself [28].

3. Experimentation

First, various pre-processing options that were experimented with are summarised. Following this the training results are benchmarked to other approaches in the literature. The text deconvolution methodology is then presented. Finally, some examples of the capability of the system to explain sentiment classification for various unseen test set examples are presented.

3.1. Pre-processing Text

3.1.1. Stop-words

Removing stop-words is a commonly used method to remove the words that would have little to no impact on the classification of a sentence. Removing words such as ’I’, ’the’, ’and’, etc., significantly decreases training and inference time. However, this method made no significant difference for this dataset, and for the task of text deconvolution we would require the original composition of the sentence.

3.1.2. Stemming

This method is used to decrease the vocabulary length of a dataset by mapping similar words such as ’fright’, ’frightened’, ’frightening’ to a same word ’fright’. However adding this modification to our pre-processing step resulted in a decrease in test accuracy. This could be blamed on the nature and perhaps the limited size of the dataset.

3.1.3. Using pre-trained word embeddings

For completeness, we experimented with using pre-trained word embedding weights for the embedding layer. To implement this approach we used ’GloVe’ embeddings [26], which are pre-trained word embeddings computed on the 2014 dump of the English Wikipedia, containing a vocabulary size of 400,000 words.

3.2. Benchmarking

For benchmarking purposes, many model architectures were compared, including recent CNN, RNN and combinations of both (that were implemented by us and found to be consistent with results reported in [29]), as well as more traditional Naive Bayes baseline methods reported in [30] and [31]. The same input pipeline and parameters were maintained in an effort to compare like-for-like based on test accuracy. Table 1 summarises the accuracies for this dataset obtained by the different models reported in the literature.

The baselines NB and BiNB are Naive Bayes classifiers with, respectively, unigram features and unigram and bigram features. RECNTN [30] is a recursive neural network with a tensor-based feature function, which relies on external structural features given by a parse tree. DCNN [31] is an early convolutional approach that utilises dynamic k-max pooling where k is determined by the sentence length. As can be seen from Table 1, the best architecture (our 1DCNN with embedding layer learned from scratch) achieved a 0.905 test accuracy evaluated on 25,000 test reviews. This approach gave slightly better (if not statistically significantly better) results than the pre-trained fine-tuned embedding layer. Training was performed using a single NVIDIA GeForce GTX 1080 Ti with 12GB of VRAM. The result shows comparable if slightly less accuracy than the state of the art [32]. However, the main contribution of this paper is in the explainability of the inferencing which we will now demonstrate.

Table 1: *Comparison of various architecture approaches*

Model	Accuracy
NB [30]	0.818
BiNB [30]	0.831
RECNTN [30]	0.854
DCNN [31]	0.868
RNN	0.880
Bi-Directional LSTM	0.881
CNN	0.895
CNN+LSTM	0.896
RCNN [29]	0.900
RCNN-HW [29]	0.903
1DCNN (pre-trained embedding)	**0.903**
1DCNN (learned embedding)	**0.905**
ULMFiT [32]	0.950

3.3. 1-D Deconvolution by Occlusion

Deconvolution by occlusion was originally proposed for image classification problems to identify what part of the input image the network looks at to support the output that it predicts [7]. Using this method, we can tell why the network classifies what it classifies, and if the network has actually trained to identify and distinguish the unique features corresponding to each class. In this paper, we propose the application of the same approach but for the text classification problem. The text deconvolution by occlusion method can be used to visualize the impact of individual words on the final prediction made by the model, see Figure 2 for an example of the output.

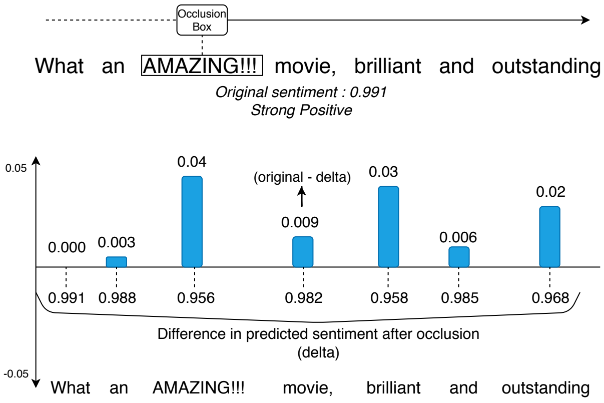


Figure 2: *Text Deconvolution with Occlusion*

The idea behind this approach is to successively occlude each element (word) present in the input sequence, and monitor the difference (delta) between the classification of the original input sequence and the prediction after masking the word. We iterate this process for all the individual words in the sentence, and monitor the fluctuation in the sentiment classification. We apply this method over the pre-processed text which have had any punctuation or white spaces already filtered out. In this way, this method is agnostic to the architecture of the model itself, since it simply modifies the input data in order to determine the effect of the modification on the output.

An alternative approach to text deconvolution in the literature is termed Text Deconvolution Saliancy (TDS) [33]. This approach is similar to activation map approaches in 2-D CNNs [7] and Layer-wise Relevance Propagation (LRP) [34]. However, like 2-D convolution activation map-based approaches, this approach requires configuration for the particular architecture under question.

3.4. Sentiment Inferencing Explained

The text deconvolution by occlusion method will now be explained; Figure 3 shows the input pipeline of the inference.

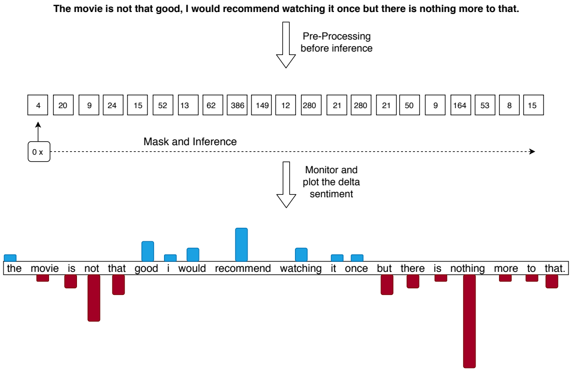
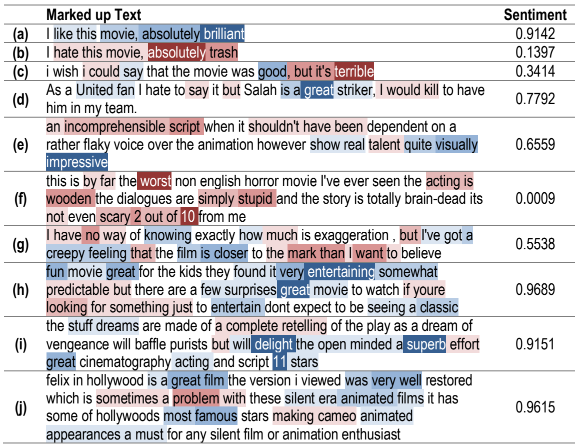

The movie is not that good, I would recommend watching it once but there is nothing more to that.
Pre-Processing before inference
4
20
9
24
15
52
13
62
386
149
12
280
21
280
21
50
9
164
53
8
15
0 x
Mask and Inference
Monitor and plot the delta sentiment
the
movie
is
not
that
good
i
would
recommend
watching
it
once
but
there
is
nothing
more
to
that.

Figure 3: *Text Deconvolution Explained*

Given an input sentence, we first need to know the sentiment of the original sentence, then we mask a single number (word) by multiplying it by 0 in the word index sequence generated after the pre-processing stage. By turning an element in the sequence into 0, the network only considers the rest of the words when making the classification. The new masked sequence is used for inference and we plot the difference between the original sentiment classification and the new occluded sentiment produced by inferencing each masked word index sequence. This process is repeated for each and every integer (corresponding to a word) in the sequence, and the plotted output gives a visualisation of the impact of each word within the input sentence, and thus contributes towards the explainability of the model's prediction.

Figure 4 shows the deconvolution at work on a few chosen reviews with their corresponding sentiment, (a) and (b) are simple reviews that contradict each-other with (a) being highly positive and (b) highly negative. These two examples demonstrate the ability of the 1D CNN to learn the context that the words are within, in (a) 'absolutely' is positive because of its relationship with 'brilliant', but in (b) 'absolutely' is negative because of the negative context of the rest of the sentence. (c) shows negation in a sentence, and illustrates that the model is looking at the sentence as a whole and not simply attributing sentiment to individual words. (d) is a positive sentence but with strong negative words like 'hate' and 'kill'. However, the model overlooks those words and focuses on the over all sentiment of the sentence predicting it as positive. This text example is the only one manufactured by the authors as an antogonistic attempt to test the model, all the other reviews are taken from the IMDB test set. Similar behavior can be observed in (e) where the overall negativity of the sentence is overwhelmed by the positive phrase 'quite visually impressive'. (f) is the most negative out of all reviews, here the model demonstrates its ability to learn from the data. The IMDB dataset includes the rating of the movie: and the user's review includes '2 out of 10', which shares the same negativity as the word 'worst' within the sentence. Similar to this we have another review (i) which is positive and the model predicts it not just because of the positive words but also because it has learned the significance of the numerical rating '11 stars'. In (g) it is difficult for a human to determine whether the review is positive or negative and this is reflected rightly in the model's neutral classification. (h) on the other hand is rightly classified as a highly positive review, despite some undermining negative phrases. Similarly, in (j) a positive review is correctly predicted despite some negative words that have been correctly put in the context of the sentence.



	Marked up Text	Sentiment
(a)	I like this movie, absolutely brilliant	0.9142
(b)	I hate this movie, absolutely trash	0.1397
(c)	i wish i could say that the movie was good, but it's terrible	0.3414
(d)	As a United fan I hate to say it but Salah is a great striker, I would kill to have him in my team.	0.7792
(e)	an incomprehensible script when it shouldn't have been dependent on a rather flaky voice over the animation however show real talent quite visually impressive	0.6559
(f)	this is by far the worst non english horror movie I've ever seen the acting is wooden the dialogues are simply stupid and the story is totally brain-dead its not even scary 2 out of 10 from me	0.0009
(g)	I have no way of knowing exactly how much is exaggeration , but I've got a creepy feeling that the film is closer to the mark than I want to believe	0.5538
(h)	fun movie great for the kids they found it very entertaining somewhat predictable but there are a few surprises great movie to watch if youre looking for something just to entertain dont expect to be seeing a classic	0.9689
(i)	the stuff dreams are made of a complete retelling of the play as a dream of vengeance will baffle purists but will delight the open minded a superb effort great cinematography acting and script 11 stars	0.9151
(j)	felix in hollywood is a great film the version i viewed was very well restored which is sometimes a problem with these silent era animated films it has some of hollywoods most famous stars making cameo animated appearances a must for any silent film or animation enthusiast	0.9615

Figure 4: *Visualisation of Deconvolution*

These results illustrate how well the trained model generalises to unseen data. The diagnostic capability of the approach explains how words from candidate sentences can be taken into context by the resulting sentiment classification.

4. Conclusions

In this paper, we have demonstrated how text deconvolution by occlusion can explain how 1-D CNNs automate classifications, providing an important diagnostic tool for debugging misclassifications and in turn improving training data and network accuracy. Unlike other approaches to deconvolution, for example Text Deconvolution Saliency [33], which relies on activation map processing, this method is completely independent of the model's architecture. For text classification systems applied to autonomous decision making, this approach could be vital for justifying how decisions are made.

5. Acknowledgements

This work is partially funded by the European Commission H2020 SC1-PM15 program under RIA grant 769872, and by Innovate UK grant 20923 'Automation and Transparency across Financial and Legal services'.

6. References

[1] "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," *Official Journal of the European Union*, vol. L119, pp. 1–88, May 2016. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC

[2] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: automated decisions and the gdpr," *Harvard Journal of Law & Technology*, vol. 31, no. 2, p. 2018, 2017.

[3] T. Z. Zarsky, "Incompatible: the gdpr in the age of big data," *Seton Hall L. Rev.*, vol. 47, p. 995, 2016.

[4] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining Explanations: An Overview of Interpretability of Machine Learning," *arXiv e-prints*, p. arXiv:1806.00069, May 2018.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXiv e-prints*, p. arXiv:1409.0473, Sep 2014.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *arXiv e-prints*, p. arXiv:1706.03762, Jun 2017.

[7] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[8] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car," *arXiv e-prints*, p. arXiv:1704.07911, Apr 2017.

[9] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," *arXiv e-prints*, p. arXiv:1502.03044, Feb 2015.

[10] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-CAM: Why did you say that?" *arXiv e-prints*, p. arXiv:1611.07450, Nov 2016.

[11] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the Rare Word Problem in Neural Machine Translation," *arXiv e-prints*, p. arXiv:1410.8206, Oct 2014.

[12] Z. Wood-Doughty, N. Andrews, and M. Dredze, "Convolutions are all you need (for classifying character sequences)," in *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 208–213. [Online]. Available: https://www.aclweb.org/anthology/W18-6127

[13] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[14] F. Malik and F. Malik, "Processing data to improve machine learning models accuracy," Nov 2018. [Online]. Available: https://medium.com/fintechexplained/processing-data-to-improve-machine-learning-models-accuracy-de17c655dc8e

[15] L. Karttunen, J.-P. Chanod, G. Grefenstette, and A. Schille, "Regular expressions for language engineering," *Natural Language Engineering*, vol. 2, no. 4, pp. 305–328, 1996.

[16] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[19] H. T. Le, C. Cerisara, and A. Denis, "Do convolutional networks need to be deep for text classification?" in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[20] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," *arXiv preprint arXiv:1705.03122*, 2017.

[21] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1480–1489. [Online]. Available: https://www.aclweb.org/anthology/N16-1174

[22] S. Gao, A. Ramanathan, and G. D. Tourassi, "Hierarchical convolutional attention networks for text classification," in *Rep4NLP@ACL*, 2018.

[23] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[24] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[28] D. López-Sánchez, J. R. Herrero, A. G. Arrieta, and J. M. Corchado, "Hybridizing metric learning and case-based reasoning for adaptable clickbait detection," *Applied Intelligence*, pp. 1–16, 2017.

[29] Y. Wen, W. Zhang, R. Luo, and J. Wang, "Learning text representation using recurrent convolutional neural network with highway layers," *arXiv e-prints*, p. arXiv:1606.06905, Jun 2016.

[30] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. [Online]. Available: https://www.aclweb.org/anthology/D13-1170

[31] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[32] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," *arXiv e-prints*, p. arXiv:1801.06146, Jan 2018.

[33] L. Vanni, M. Ducoffe, D. Mayaffre, F. Precioso, D. Longrée, V. Elango, N. Santos, J. Gonzalez, L. Galdo, and C. Aguilar, "Text deconvolution saliency (tds): a deep tool box for linguistic analysis," in *56th Annual Meeting of the Association for Computational Linguistics*, 2018.

[34] L. Arras, F. Horn, G. Montavon, K.-R. Müller, and W. Samek, "" what is relevant in a text document?": An interpretable machine learning approach," *PloS one*, vol. 12, no. 8, p. e0181142, 2017.