COMP551 Mini-Project 4

Akshal Aniche, Dylan Sandfelder, Jacob Sanz-Robinson

April 2019

1 Abstract

As companies invest increasingly more money into maximizing profit and customer satisfaction, it becomes necessary to build robust machine learning models that perform sentiment analysis to be able to assess reviews and critiques without wasting human resources. In this paper, we attempt to recreate one of the models presented in "Convolutional Neural Networks for Sentence Classification" (Kim, 2014), and implement a simple model of our own to beat the baselines presented in the paper. We worked with three of the datasets used by Kim to create binary classification models. We implemented a convolutional neural network using word2vec embeddings to create features, and a stacking of logistic regression, linear SVM and k nearest neighbours using trigram TF-IDF features. Although our models performed very well compared to the previous state of the art baselines in Kim (2014), they did not achieve as high accuracy as the complex CNNs presented by Kim.

2 Introduction

A machine capable of performing sentence classification tasks such as sentiment analysis is an exciting notion for various stakeholders for a number of reasons. Obvious benefits to companies include being able to use that information to assess their public image and product responses. Moreover, sentiment analysis can also be useful for mental health monitoring systems which can benefit communities at large, as seen in papers such as Wang et al. (2013), Zucco et al. (2017), and De Choudhury et al. (2013) where depression is detected and monitored using data from social media. It can be used to learn more about the nature of mental health conditions and observing the effects they have on people's lifestyle and interactions through social media. It is clearly a technology with the capacity of revolutionizing our quality of life for the better, which is why we decided on choosing to work on this domain as our final COMP 551 project. The paper we chose for this project, "Convolutional Neural Networks for Sentence Classification" (Kim, 2014) attempts to beat state of the art baselines in sentence classification by using convolutional neural networks.

As such, one of the objectives of this project is to implement and reproduce some of the baselines presented by Kim, and implement an approach using our own convolutional neural networks to see whether extensive hyper parameter optimization is needed to achieve the results obtained in the original research paper. The other objective is to approximate these baselines using an ensembling approach, stacking many of the simpler Machine Learning models we saw in the course.

3 Related works

Text classification and sentiment analysis are fields of machine learning and natural language processing that have been active for decades. In this section of the report, the paper we chose will be summarized, critiqued, and compared with several results and approaches featured in literature dealing with similar subject matters. We also provide a brief review of the field in a chronological fashion.

3.1 Previous work in sentiment classification tasks

First we will examine some of the work that came before Kim (2014). Turney (2002) shows that an unsupervised classifier can achieve 74% accuracy on the task of rating reviews positively or negatively. Since then, the field has become increasingly popular. Zhang & Yang (2003) showed that Support Vector Machines (SVM) were more robust than logistic and ridge regressions in text classification purposes, specially under noisy conditions (such as incorrect labels), or datasets with unbalanced classes, both of which are issues pertinent to this project, and StanfordNLP (2009) highlights various works from the early 2000's showing the successes of SVMs used in conjunction with ensemble methods such as boosting.

In 2008, there was a rise in popularity in the use of Naive Bayes classifiers, which are typically successful in text and sentiment classification systems (Brmez, 2016). In 2012, the seminal "Bigrams and Baselines" paper showed that bigrams improve the performance of sentiment classification systems, having greater sentimental effects than individual words when used as features (Wang & Manning).

3.2 Sentiment classification and ethics

As we have introduced, sentiment classification holds high stakes for profit-based companies. The ability to automatize sentiment classification can be combined with classic text mining strategies to analyze the public's opinion on particular products and trends, with very little human effort. For example, movie box office reviews play an important role in determining box office revenue (Duan et al., 2008). In Kim (2014), five of the seven datasets used are reviews of commercial products such as goods or movies. With this information, companies can adjust their future endeavour to maximize their profit, necessary to survive in a highly capitalistic market. Taking into account the financial impact of such reviews, we have to consider the possible bias in the data used for these tasks.

Let us consider the movie industry in North America. There is an imbalance in the representation in movies between minorities and majority groups. Smith et al. (2016) found that of the top 100 films in 2014, 73.1% of the characters were white, as opposed to 12.5% Black and <1% Indigenous American or Native Hawaiian/Pacific Islander. Out of 4610 speaking characters, 19 were represented as lesbian, gay or bisexual, and none were transgender. Only 30.2% of speaking characters of the top 700-grossing movies of 2014 were female. This inherent imbalance would skew the results of classifying movies.

Furthermore, in American society, systemic oppression against minorities is omnipresent and is ingrained into individuals. This bias is reflected into assessments of work that includes such minorities. For example, Fowdur et al. (2012) found that "ratings for movies with a Black lead actor and all white supporting cast are approximately 6 percent lower than for other racial compositions." This leads to racial bias being learned by classifiers, later wrongly classifying movies featuring Black leads as unsuccessful. A movie featuring minority leads that is classified as unsuccessful would discourage studios from investing into such movies, because of the lack of historical data that would classify this datapoint as an anomaly, fueling the vicious circle that prevents marginalized people from accessing work opportunities in the industry.

Sentiment analysis can also be used to study social and human trends. A lot of work has been done to analyze the rise of white supremacy on online, far-right forums. Figea et al. (2016) created a model to measure online affects in such forums, and Sureka & Agarwal (2014) attempts to classify hate in tweets, to find tweets that would violate the Rules and Regulations of the Terms and Services of Twitter. This work is crucial in order to be aware of the rise of fascism and the role of social media in it. However, one nuance that is often ignored by such models is that of defense speech and reappropriated slurs, classifying text from marginalized individuals as hate speech without considering the power dynamics inherent to oppressive structures. For instance, there is a documented pattern of Facebook restricting Black activists' accounts for defending themselves against white supremacist attacks (Urana-Ravelo, 2016).

3.3 Critical summary of Kim (2014)

The paper we chose for this project was published in 2014, and takes a slightly different approach from its predecessors. Kim uses a simple Convolutional Neural Network (CNN) to classify sentences in 7 datasets. On four of these datasets, Kim beat state of the art baselines. In terms of the implementation, the paper uses

padding, ensuring the input sentences are of the same length, and then maps the words in the resulting padded sentences to word embeddings using word2vec (Mikolov et al., 2013), which produces a 300-dimensional vector describing the closeness of words in the corpus (based on a pre-training on words from Google News). At the time, this was a novel tool to use as a feature, and as the results show, an important component in the success of the approach.

Their main CNN model employs a convolutional layer with multiple filter widths and feature maps, followed by a max-over-time pooling operation over the feature maps. The pooling results from different layers are then concatenated and fed into a fully connected layer with softmax activations, which outputs the probability of each label for the particular example. For regularization, dropout is used on the penultimate layer.

Kim's approach trains the CNNs with a batch size of 50 and the AdaDelta optimizer, which we will not delve into in this report. The convolutional layers all feature RELU activation functions, "filter windows of 3, 4, 5 with 100 feature maps each", a dropout rate of 0.5, and L2 constraint (gradient clipping) of 3, selected using grid search.

Kim trains 4 variations of CNNs. CNN-rand uses initially randomized words that are updated in training, CNN-static uses the static pre-trained word2vec embeddings while learning the other parameters, CNN-non-static updates the word vectors during training, and CNN-multichannel uses two sets of vectors (one static, and the other non-static). In terms of results, it can be seen that the pre-trained vectors work well, and that non-static approaches work better then their static counterparts. The multichannel model doesn't succeed in preventing overfitting as was initially hypothesized, and the 0.5 dropout was an effective regularization technique. In terms of beating benchmarks, the CNNs (with the exception of CNN-rand) all outperformed the previous literature on at least one dataset. Benchmarks were set for 4 of the 7 featured datasets, at 81.5% accuracy for the MR dataset, 88.1% for SST-2, 85.0% for CR, and 89.6% on MPQA. More information on these datasets is given in the following section of the report. Most interestingly, at the time the paper was published, the mechanism through which a CNN could "understand" text had not yet been thoroughly analyzed, and it prompted many further studies on the topic, one example being Jacovi et al. (2018), which concludes, among other things, that max-pooling over time distinguishes useful features, and that linguistic patterns are captured by the filters when n-grams are fed through them.

While it is clear that Kim (2014) shows an effective and novel approach to the decade-old problem, there is one detail we take issue with in the paper. The experiments featured in the paper are evaluated by their accuracy. However, the samples in the datasets are not uniformly distributed across all of the classes, even more markedly so for the binary labels. For example the binary MPQA dataset, 68% of the data is labelled 'negative'. As we know, accuracy is not the best metric to evaluate model performances on unbalanced datasets. (On a dataset labelled 99% positive, a model can achieve 99% accuracy by always predicting the positive label). It's worth pointing out that the datasets are rather small compared to others we have seen used in sentence classification papers.

Since then, further improvements and new approaches have been made in the field. The newer approaches can achieve upward of 94% accuracy on numerous review datasets, as is the case with neural networks with block-sparse weights (Gray et al., 2017), and transfer learning methods for text classification (Howard & Ruder, 2018). The objective of task 1 of the project, however, is to surpass Kim's papers baselines using the simpler methods seen in class, so we will not delve into these newer method any further.

4 Dataset and experimental setup

We set out to test our baselines on three of the datasets featured in the original paper. The datasets we chose were those in which the original paper was successful at outperforming previous literature and have binary labels, namely the Customer Review (CR) (Hu & Liu, 2004), Multi-Perspective Question Answering (MPQA) (Wiebe et al., 2005), and the binary version of the Stanford Sentiment Treebank (SST-2) datasets (Socher et al., 2013). The original paper also outperformed previous literature on the Movie Review (MR) dataset, but we decided to omit this dataset from our trials as it is a subset of SST-2.

The three datasets consist of phrases and their binary labels. For the CR dataset, the objective is to predict positive or negative sentiment given phrases extracted from reviews of different products. It contains

a total of 3744 phrases. For the SST-2 and MPQA datasets, the objective is to predict positive or negative sentiment given phrases from movie reviews. SST-2 contains 10662 examples, and MPQA contains 10605 examples.

Each of the datasets was split in a training set and a validation set in a 80%-20% ratio using the SciKit Learn package Pedregosa et al. (2011). The training was performed using Keras for TensorFlow Chollet et al. (2015) in Python 3, and was run on on a Google Colab Notebook using Google's Tesla K80 GPU.

We used different data preprocessing techniques for the two different approaches we attempted, which will be detailed further in the following section.

5 Proposed approach

In this project, we ran a series of convolutional neural networks along with other more traditional supervised models to achieve a baseline score that would rival the results in Kim (2014).

5.1 Convolution neural networks

We designed our CNN based on the ones described in Kim's paper. The CNNs used in the paper are subject to a rigorous amount of parameter optimization which we opted not to apply to our CNN, both due to time constaints, and due to the fact that our main interest lies in investigating whether simple, non-optimized CNNs are competitive with the CNNs featured in Kim's paper, and with other simpler machine learning models.

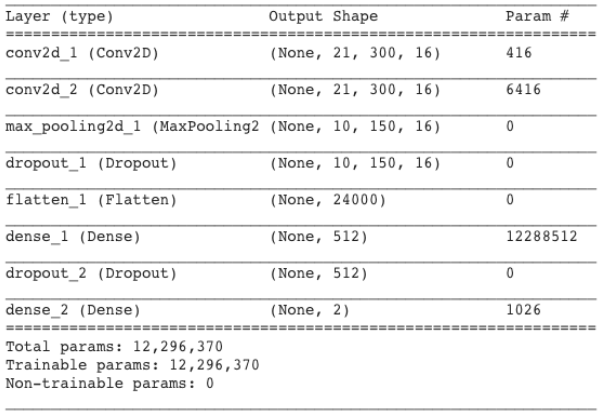
Before the data could be fed into our CNN, it was shaped into a format that would both be accepted by our CNN and would be a feature conducive to making predictions. In this respect, we emulated the paper as best we could. Each sentence is first turned from a series of words to a series of 300-dimensional vectors obtained from the word2vec embeddings. To standardize the length of these vector lists, each list is then padded with 300-length zero vectors, until they are all as long as the longest sentence in that particular dataset. Finally, the entire dataset consisting of the lists of vectors is turned into one large NumPy array, which is in a format that can be fed into our CNN.

The CNN we built is relatively simple, and is constructed in semblance to the CNNs in Kim's paper. The paper does not go to great lengths to describe the details of their CNN implementation, but it does say that they are constructed using one convolutional section and one fully connected section, with a maxpool and dropout in between. This is the design that we chose to use. In particular, our CNN uses an input convolutional layer with a filter size of 16 followed by another convolutional layer with a filter size of 16. This is immediately followed by a maxpool and a dropout with a parameter of 0.5 before leading into our fully connected section. This section is made up of dense layer that uses arectified linear unit activation, followed by a dropout of 0.5, and finally a dense output layer that uses softmax activation (which is the same activation used in the output layer of the paper we are trying to emulate). A summary of the model, generated using the Keras package, can be seen in Figure 1.

5.2 Ensembling method

We found multiple papers highlighting the success of ensembles in both sentiment and text classification Whitehead & Yaeger (2008), Liu et al. (2003). This led us to consider an approach where we combined a number of the simple models seen in class as a simple and reliable way of obtaining competitive results. We experimented with a few simple linear classification methods to reach high accuracy. All of these models were implemented using the Python 3 ScikitLearn package (Pedregosa et al., 2011).

Before creating features for the data, we filtered the text to remove non-alphabetical tokens, and converted the text to lowercase. Then the data was processed using the ScikitLearn TfIdfVectorizer class. This generates TF-IDF features from all the monograms, bigrams and trigrams in the training set, and then normalizes the features to the interval [0,1]



Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 21, 300, 16)	416
conv2d_2 (Conv2D)	(None, 21, 300, 16)	6416
max_pooling2d_1 (MaxPooling2	(None, 10, 150, 16)	0
dropout_1 (Dropout)	(None, 10, 150, 16)	0
flatten_1 (Flatten)	(None, 24000)	0
dense_1 (Dense)	(None, 512)	12288512
dropout_2 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 2)	1026

Total params: 12,296,370
Trainable params: 12,296,370
Non-trainable params: 0

Figure 1: Summary of our CNN

The first of our models uses logistic regression to classify the datasets, with the L-BFGS solver and L2 regularization ($\lambda = \frac{1}{80}$). The main motivation for choosing to run experiments on a logistic regression model comes from previous work showing that when used on large sparse datasets with binary outputs (like our datasets), they can equal or improve upon the performance of models such as Naive Bayes and SVMs Komarek & Moore (2003), Zhang & Yang (2003), K Sarma & Sethares (2018). Structurally, the capacity of logistic regression to support a rich and large feature representation, along with the fact that it doesn't assume a linear trend between the independent and dependent variables indicate it is well suited for text classification purposes.

Our second model is a linear soft SVM, with L2 regularization, a tolerance of 0.01 on the F1 measure and penalty parameter $C = 3$ on the error term. In the related works we mentioned that SVMs were comparatively robust to noise and imbalanced datasets, as pointed out in Zhang & Yang (2003). Seeing how out of the whole bag of words of each corpus, we expect relatively few of the words to contribute significantly to the sentiment (the rest of them being noise), and how the datasets we used are not evenly split, this seemed like a good choice of classifier to include in our ensemble.

Our third model is a k-Nearest Neighbours model, with $k = 22$, using Euclidian distance for weights, which was a model we also saw used in literature as an effective text and sentiment classifier Bruno et al. (2013), Rezwanul et al. (2017).

Each of these models has hyperparameters that were optimized using grid search with the CR dataset and F1 measure as score. The F1 measure takes into account both the precision and the recall of the model, which provides a more reliable scoring of the models on unbalanced data than accuracy.

Finally, all of these models were stacked using a voting classifier, with hard voting.

6 Results and discussion

6.1 Convolutional neural networks

Each CNN (one per dataset) was run for 50 epochs to make sure the models were fully complete before reading their final results.

Our CNNs started overfitting early during training, after only a few epochs. In part this is to be expected as the datasets are quite small compared to other datasets we have seen in the field, but it also shows how powerful of a tool CNNs can be when used for sentiment analysis and text classification. This fact can clearly be evidenced in the graph of the training and validation accuracy levels in the MPQA dataset (Figure 2).

6.2 Ensembling method

After training the simple, non-neural network, models on each dataset, we measured their performance on the test set with the F1 measure, presented in Table 1.

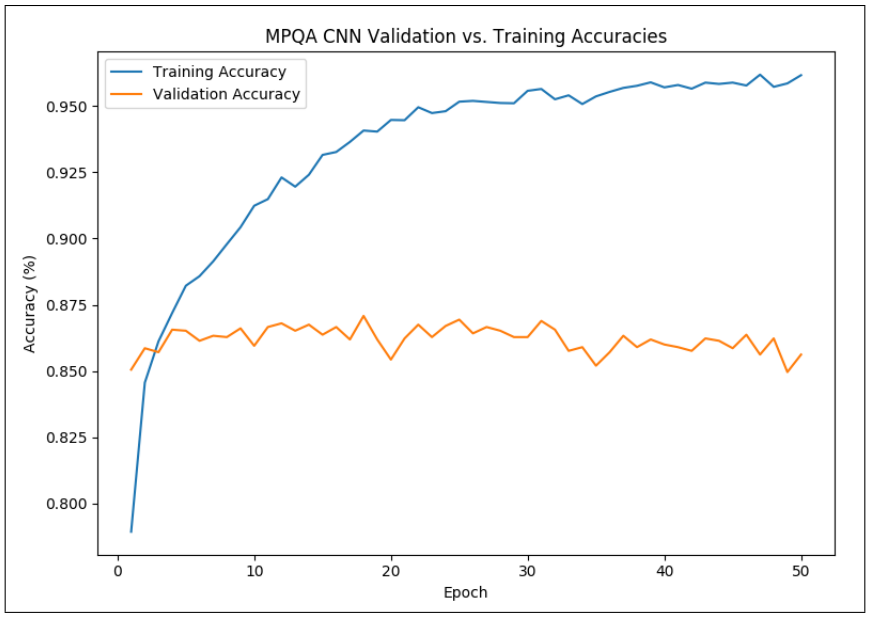


Figure 2: The evolution of the training and validation accuracy results of the CNN for the MPQA dataset.

Table 1: Comparison of the F1 scores of the models

Dataset	**Logistic Regression**	**Linear SVM**	**k-Nearest Neighbours**	**Stacked model**
CR	0.85132	0.84696	0.81437	0.85597
MPQA	0.77707	0.76279	0.69257	0.77460
SST2	0.80755	0.80974	0.79147	0.80686

As we can see, our different individual models achieve high performance of the datasets, and this performance is slightly improved by stacking the models with hard voting.

This justifies our intuition that the ensemble model would perform very highly.

6.3 Comparing to Kim (2014)

We can now compare our models to the baselines presented by Kim (2014). To be able to do that, we computed the accuracy of the model on the three datasets, even though as we discussed, it is not a reliable measure of performance for this task.

Our CNN results should also be compared individually to the "CNN-static" model's results featured in the paper, which it is most similar to.

Results are shown in Table 2.

Table 2: Accuracy achieved by Kim's models and our models on the CR, MPQA and SST2 datasets

Dataset	**Kim's Best CNN**	**CNN-static**	**Our CNN**	**Our ensemble**
CR	0.850	0.847	0.763	0.815
MPQA	0.896	0.896	0.871	0.866
SST2	0.881	0.868	0.745	0.799

Our CNN results varied in accuracy between the datasets. The highest accuracy we achieved was of 87.1% for the MPQA dataset, which came close to the results of Kim's paper for the MPQA. The MPQA dataset lent itself very well to this task, as even our simple CNN outperformed most of the previous baselines

mentioned in Kim's paper, which used a variety of supervised models ranging from SVM variations to Conditional Random Fields Kim (2014). For the CR and SST-2 datasets, our CNN didn't reach the benchmarks set by Kim's paper, and were outperformed by roughly 10 percentage points.

On the other hand, our ensemble model was consistent and had results that were expected, even beating most of the previous state of art baselines on the CR and MPQA datasets. Even though it didn't beat Kim's CNNs, it did beat our simple CNN for the CR and SST2 dataset.

All in all, it is fair to say that the lengths Kim went to to optimize the hyper parameters of his CNNs do appear to have paid off. Using non-static word vectors to create features for CNNs created a robust model that, unlike our models that equal the performance of previous baselines, performs noticeably better.

7 Conclusion

In this project we attempted to reproduce the results shown in Kim's 2014 paper on sentence classification, and implement our own ensemble model (using stacking) to approximate these baselines using simple classifiers. Running our experiments on three binary datasets, we found that implementing a CNN only approximated the paper satisfactorily for the MPQA dataset (which was within 2 percentage points of the accuracy of the corresponding model from the paper). Implementing our own ensemble which used stacking with a Logistic Regression, a Linear SVM, and a k-NN model, we achieved similar, more consistent results, very close to the previous state of the art baselines.

While our implementations were not able to surpass Kim's models, we were pleasantly surprised to find that our ensembling approaches, while based on simple models which have been largely untouched in decades, perform adequately in terms of accuracy. The fact that these models performed similarly to our CNN, and to some of the earlier baselines mentioned in Kim's paper goes to show that unless the data, parameters, and hyperparameters are extremely finely tuned, or extremely complex features like non-static word embeddings are used, simple approaches can be just as effective as approaches based on more complex models.

Finally, we feel the models we built, and even models like Kim's CNNs don't capture the full complexity and nuance of human language. While the newer papers mentioned in the Related Works section achieve high accuracy by using larger datasets and acquiring higher level representations of data through deep learning, it would seem like the future of the field lies more towards the development of features capable of constructing a more detailed overview of sentiment. Features that take into account aspects of sentiment such as intensity Thet et al. (2010), features that make use of grammatical structure, and features learnt from big data, coupled with the use of domain specific corpuses are the main possibilities for exciting developments and new applications in the fields of sentence and sentiment classification Liang et al. (2017). Of course, keeping in mind the ethical implications of models built is always necessary when working with high stakes concepts. It is even useful to think about including fairness and justice into our model, similarly to the work done in Zemel et al. (2013) to learn fair representation.

References

Brmez, S. (2016). Analysis of complex sentiment on social networks.
URL https://pdfs.semanticscholar.org/a943/8f3a80e4ab5ab69c107da23c85b253f6fe5d.pdf.

Bruno, T., Sasa, M., & Donko, D. (2013). Knn with tf-idf based framework for text categorization. vol. 69.

Chollet, F., et al. (2015). Keras. https://keras.io.

De Choudhury, M., Gamon, M., Counts, S., & Horvitz, E. (2013). Predicting depression via social media.
URL https://www.microsoft.com/en-us/research/publication/predicting-depression-via-social-media/

Duan, W., Gu, B., & Whinston, A. (2008). The dynamics of online word-of-mouth and product sales-an empirical investigation of the movie industry. *Journal of Retailing*, *84*(2), 233–242.

Figea, L., Kaati, L., & Scrivens, R. (2016). Measuring online affects in a white supremacy forum. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, (pp. 85–90).

Fowdur, L., Kadiyali, V., & Prince, J. (2012). Racial bias in expert quality assessment: A study of newspaper movie reviews. *Journal of Economic Behavior & Organization*, *84*(1), 292–307.

Gray, S., Radford, A., & Kingma, D. P. (2017). Gpu kernels for block-sparse weights.

Howard, J., & Ruder, S. (2018). Fine-tuned language models for text classification. *CoRR*, *abs/1801.06146*.
URL http://arxiv.org/abs/1801.06146

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 168–177). ACM.

Jacovi, A., Shalom, O. S., & Goldberg, Y. (2018). Understanding convolutional neural networks for text classification. *CoRR*, *abs/1809.08037*.
URL http://arxiv.org/abs/1809.08037

K Sarma, P., & Sethares, W. (2018). Simple algorithms for sentiment analysis on sentiment rich, data poor domains. In *Proceedings of the 27th International Conference on Computational Linguistics*, (pp. 3424–3435). Santa Fe, New Mexico, USA: Association for Computational Linguistics.
URL https://www.aclweb.org/anthology/C18-1290

Kim, Y. (2014). Convolutional neural networks for sentence classification. *CoRR*, *abs/1408.5882*.
URL http://arxiv.org/abs/1408.5882

Komarek, P., & Moore, A. W. (2003). Fast robust logistic regression for large sparse datasets with binary outputs. In *AISTATS*.

Liang, H., Sun, X., Yunlei, S., & Gao, Y. (2017). Text feature extraction based on deep learning: a review. *EURASIP Journal on Wireless Communications and Networking*, *2017*.

Liu, Y., Carbonell, J., & Jin, R. (2003). A new pairwise ensemble approach for text classification. (pp. 277–288).

Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, *abs/1301.3781*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Rezwanul, M., Ali, A., & Rahman, A. (2017). Sentiment analysis on twitter data using knn and svm. *International Journal of Advanced Computer Science and Applications*, *8*.

Smith, S. L., Choueiti, M., Pieper, K., Gillig, T., Lee, C., & DeLuca, D. (2016). Media, diversity, & social change initiative.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, (pp. 1631–1642).

StanfordNLP (2009). Stanford: Support vector machines.
URL `https://nlp.stanford.edu/IR-book/html/htmledition/references-and-further-reading-15.html`

Sureka, A., & Agarwal, S. (2014). Learning to classify hate and extremism promoting tweets. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, (pp. 320–320). IEEE.

Thet, T. T., Na, J.-C., & Khoo, C. S. (2010). Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of Information Science*, *36*(6), 823–848.
URL `https://doi.org/10.1177/0165551510388123`

Turney, P. D. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. *CoRR*, *cs.LG/0212032*.
URL `http://arxiv.org/abs/cs.LG/0212032`

Urana-Ravelo, B. L. (2016). Facebook hates Black people. *Medium*.
URL `https://medium.com/indian-thoughts/facebook-hates-black-people-ff2579f18b03`

Wang, S., & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, (pp. 90–94). Stroudsburg, PA, USA: Association for Computational Linguistics.
URL `http://dl.acm.org/citation.cfm?id=2390665.2390688`

Wang, X., Zhang, C., Ji, Y., Sun, L., Wu, L., & Bao, Z. (2013). A depression detection model based on sentiment analysis in micro-blog social network. In J. Li, L. Cao, C. Wang, K. C. Tan, B. Liu, J. Pei, & V. S. Tseng (Eds.) *Trends and Applications in Knowledge Discovery and Data Mining*, (pp. 201–213). Berlin, Heidelberg: Springer Berlin Heidelberg.

Whitehead, M., & Yaeger, L. (2008). Sentiment mining using ensemble classification models. (pp. 509–514).

Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, *39*(2-3), 165–210.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., & Dwork, C. (2013). Learning fair representations. In *International Conference on Machine Learning*, (pp. 325–333).

Zhang, J., & Yang, Y. (2003). Robustness of regularized linear classification methods in text categorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, (pp. 190–197). New York, NY, USA: ACM.
URL `http://doi.acm.org/10.1145/860435.860471`

Zucco, C., Calabrese, B., & Cannataro, M. (2017). Sentiment analysis and affective computing for depression monitoring. (pp. 1988–1995).