**Hate Speech Detector**

| **Answers to the questions/ discussions in the notebook** |
| --- |
| 2.10. DISCUSS: are these hashtags making sense? should we include them as features or should we strip the # before tokenizing (that is, treat "#love" the same as "love")? why and why not?  #YOUR ANSWER HERE: People use hashtags to convey a very specific idea or context to which he/she wants to associate its tweet. So, it contains a lot of information towards the purpose of the tweet. I believe this information should be kept and in a special category to denote more importance than common words. Hashtags, on the other side, are less frequent than other words, so stripping them from the # before tokenization would reduce even more its impact. The hashtags are similar to the top words we found before. So, if one considers that the labeling is accurate, they make sense.  We could have also done some different word processing to retain the most meaning from the tweets like chat word conversion and spelling correction. As I mentioned above, a sort of hashtag to words conversion is needed too. |
| 3.8. Discuss: how does w2v calculate the similarities?  YOUR ANSWER HERE: W2v associates each word to a ‘position’ in the embedding space by learning the representation of words as vectors in the multi-dimensional embedding space. Then w2v calculates similarities between words by estimating the cosine of the ‘angle’ formed by the vector representing those two words in the multi-space of word embeddings. The idea behind this is that the closer in meaning they are, the closer to zero that angle should be (or that the projection of one word-vector on the other will be closer to its full ‘size’). By representing word as vectors, w2v can also perform vector addition and subtraction between related word-vectors and apply that space difference to other words to predict a similarly related word. |
| 3.9. Discuss: do you think Word2Vec is supervised or unsupervised ML technique?  YOUR ANSWER HERE: It is unsupervised in the sense that there are no labels in the training data and w2v ‘discovers’ the relationship between words, but there is some ‘supervision’ as the way you train the model is by predicting a word (cbow) from a given set of words, or by predicting a word’s neighbors given a word (skip gram).  For these characteristics, w2v is usually referred to as being self-supervised learning. Because it uses the implicit relationship in words to create labels for each word. |
| 4.6. Discuss the differences in performance using tf-idf vs skim-gram embeddings.  YOUR ANSWER HERE: Tf-Idf is a statistical measure that determines how significant a word is to a document in a collection of documents. These metrics are easly to calculate and thus an embedding based on them is faster to implement.  Overall scores (f1) are better for the tf idf embedding than for the skip gram embedding used in the w2v model. The precision for the skip gram embedding is very low for classifying racist/sexist tweets. |
| 4.7. Examine a few tweets where the model(s) failed.  What other features would you include in the next iteration?  YOUR ANSWER HERE: Looks like we should keep more words (maybe skip removing generic stop words) in the tweets to get a better classification. |
| 5.7. The model starts to overfit after a couple of epochs. Consider using [early stopping](https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping) to stop training when a monitored metric has stopped improving.  What can we do to tame overfitting?  YOUR ANSWER HERE: There are several actions that one can take to reduce overfitting. One is to increase the size of the training set so that the model can better predict unseen data. Use cross validation (dividing the training set in k groups and train with k-1 group and test with the remaining one, and do this for each group), this is another way ‘increasing’ the training potential of the dataset. Reduce the number of features in the data set so that the model will have less tendency to overfit, especially if the training set is limited. Remove layers or the number of nodes in the dense layers, as over complex models are more likely to overfit. Apply drop out (disconnecting some nodes, in a random pattern, while training the model) to reduce the interdependence of the nodes in the model. Do early stopping, which is to stop training the model when some particular metric ( measured on the validation set) stops to improve, and using the resulting model as |
| 6.6 Jot down your observations in explaining the model.  YOUR ANSWER HERE: LIME provides an easy method to estimate how much each word have on the final classification of the tweet. In this case it selected the 6 (or less) most important categories and calculates their impact on the classification.  Lime is a black-box explainer that allows us to explain the decisions of any classifier **on one particular example** by perturbing the input (in our case removing words from the sentence) and seeing how the prediction changes. |