

HTSPC- Hate Speech Classification

Problem statement- We are given a piece of text, which we need to classify into hate speech or not hate speech. It is a binary classification problem with labels “HOF”(0) denoting hate speech and “NOT”(1) denoting non hateful sentences. Numbers in brackets denote the class label.

- GloVe is a word vector technique that leverages both global and local statistics of a corpus in order to come up with a principled loss function which uses both these.
- Word vectors put words to a nice vector space, where similar words cluster together and different words repel.
- The advantage of GloVe is that, unlike Word2vec, GloVe does not rely just on local statistics (local context information of words), but incorporates global statistics (word co-occurrence) to obtain word vectors.
- For training purposes, we load the whole embeddings into the memory. And use the word in the training data to create weight matrix which is used to assign weights in the model layers.

create a weight matrix for words in training docs

```
In [8]: embedding_matrix = zeros((vocab_size, 100))
for word, i in t.word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector
```

- Then the trained model is used the predict labels for the test data.

define model

```
In [9]: model = Sequential()
e = Embedding(vocab_size, 100, weights=[embedding_matrix], input_length=20, trainable=True)
model.add(e)
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
```

summarize the model

```
In [11]: print(model.summary())
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
=====		
embedding_1 (Embedding)	(None, 20, 100)	2021600

flatten_1 (Flatten)	(None, 2000)	0

dense_1 (Dense)	(None, 1)	2001
=====		
Total params: 2,023,601		
Trainable params: 2,023,601		
Non-trainable params: 0		

None