FM 9528 Banking Analytics

Coursework 3

Student ID: 251121253

Word Count: 2203

Part a) Data Processing

(Please see Appendix Section: Data Processing)

Treatment Order	Elements in the Text	Treatment (FastText and GloVe)
1	• Website Links (URL)	• Directly Removed • Reason: a website link can create troubles for embedding due to its extremely long characters and various punctuations. It also does not imply any sentiment from the writers too.
2	• Emoticons	• Converted to text that represents the meaning of the emoticons • Reason: a emoticons uses combinations of punctuations to express a sentiment, which is very important for our analysis. Since FastText and GloVe cannot read punctuations, so we convert them to text to preserve the sentiment information.
3	• Emoji	• Converted to text that represents the meaning of the emojis • Reason: similar to emoticons, a Emoji has a great contribution to interpret the sentiment. Since FastText and GloVe cannot read emojis, so we convert them to text to preserve the sentiment information.
4	• Contractions (e.g. you're, haven't)	• Expand contractions to full words • Reason: FastText and Golve may not read the contractions properly, so we expand them for proper embedding
5	• Punctuations	• Removed after above treatments are done • Reason: all useful punctuations useful in interpret sentiment has been processed. The rest of them has no contribution to interpret sentiment. • "?" and "!" may have some minor use, but they emphasize more on intensity of the sentiment not sentiment itself, therefore, I did not convert them to text meaning. • FastText and GloVe does not read punctuations as well
6	• Double Spaces	• Directly Removed • Reason: no use for analysis.

Treatment Order	Elements in the Text	Treatment (FastText and GloVe)
7	• Stoping Words	• Removed only a few of them as follows: "the", "a", "an", "and", "or", "to". • Reason: removing these above stopping words does not affect the meaning of the text. However, if we remove other stopping words such as subjectives (e.g. "I", "They") and propositions (e.g. "before", "after"), it will change the meaning of the text and affect the sentiment. Therefore, I kept other stopping words.
8	• Abbreviation (e.g. lol, omg, WTF)	• Preserved in the data • Reason: Since most of the common used abbreviations such as "lol" have explanatory articles on Wikipedia or a record in Common Crawl, the embedding packages such as FastText and GloVe would capture them and give appropriate vectors to represent them.
9	• Uncased and cased words	• All text are lower cased • Reason: no significant differences in meaning for cased and uncased words in English, so transferring all text to lower case will not affect the meaning of the comment. At the same time, all texts in different forms will be treated as the same. For example: "good", "Good", and "GOOD" imply the same meaning. Some intensity of sentiments expressed through capitalization may be weakened, but does not affect the sentiment expressed through those words.

*Notice that all the treatments above are done in the treatment order so that all of the elements can be processed properly.

All the emoji & emoticon dictionary are created by Neel Shan (2019)

Each resulting comment text is in the following forms:

- A long sentence of plain text in lower cases
- No contractions and punctuations
- Some abbreviations
- Some sentences may have missing conjunction words.
- Some verbs or nouns that does not match with the sentence's flow may appear in the middle or end of the sentence. They are the results of the conversion of emojis and emoticons to indicate the sentiment.

Part b) Data Statistics

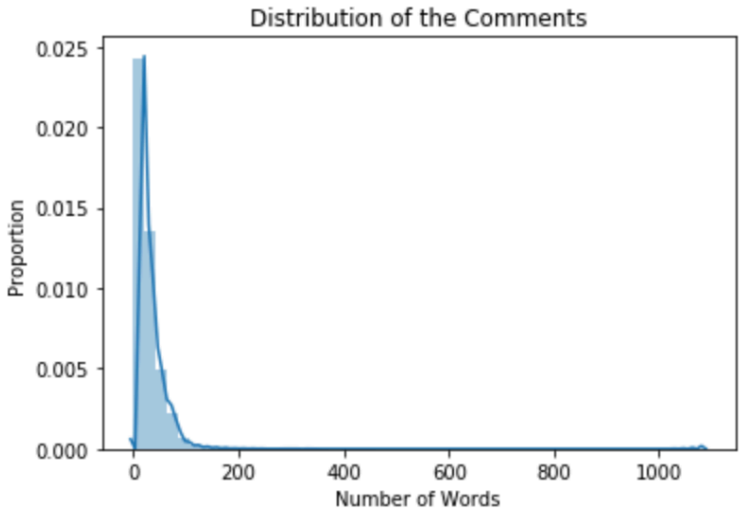
The top 5 most repeated words are as following:

Words	Repeated Times
“i”	195186
“you”	114632
“is”	104459
“it”	100900
“for”	87645

The phrase statistics :

Average Number of Words	Standard Deviation of Words	Maximum Words in a Comment	Minimum Words in a Comment
27.59	25.47	1084	1

Distribution of the Comments



The distribution of the comment has a long tail on the right side. 99% of the comments have words less than 200 words. Therefore, I chose 175 words as the cutoff level for trimming and padding. These comments are from crowdfunding websites, and the words at the beginning of the comments mostly are addressing people’s names or greetings. All the important information for interpreting the sentiments locates in the later part of the comments. Therefore, this is an important indication for using “**pre-padding**” to capture the key information.

Part c) Embedding & Neural Network

(please see details in Appendix Section: Embedding & Neural Network)

Embedding Layer:

Most of the comments in this data set are in an informal Internet language tone. Because Common Crawl is a data set that archives all websites information (including those Internet language tones), embeddings trained through Common Crawl would interpret those texts more accurately than those trained through Wikipedia, which only contains formal articles. Therefore, choosing embeddings trained through Common Crawl is most appropriate for this data set.

The following two pre-trained embedding models were used for embedding the processed data:

- FastText:
    - ✤ Pre-trained package: crawl-300d-2M-subwords.
    - ✤ This model has 2 million word vectors trained with subword information on Common Crawl.
- GloVe:
    - ✤ Pre-trained package: glove.42B.300d
    - ✤ This model has 1.9 million vocabulary, uncased, 300d vectors, trained over Common Crawl

Reason to choose these two embeddings:

With the training data stick with Common Crawl and the principle techniques fixed as the static 300d vector representation, I chose FastText and GloVe because I want to compare how much improvements can be made by the subword methodology. In theory, subwords would correct spelling mistakes and read texts more accurately. So the embeddings above are my control (GloVe) and experimental (FastText) groups. As mentioned in part b), I used “pre-padding” with a trim size of 175 to standardize the sequences of the text data.

The resulting embedding matrix is in a dimension of (# of unique words + 1) x (300)

Use Confidence Level to Sample Training Data:

The confidence level of a sentiment label plays an essential role in identifying useful data for training. If unconfident labels are directly fed into the models, it will cause bias or confusion for the model to identifying key sentiment features. Notice that the confidence level for the sentiment labels ranged from 0 to 22.29, which is really hard to interpret how confident the labels are; therefore, I rescaled the confidence level with the following linear transformation:

$$\text{New confidence level} = \frac{current - oldmin}{oldmax - oldmin} \cdot (newmax - newmin) + newmin$$

where:

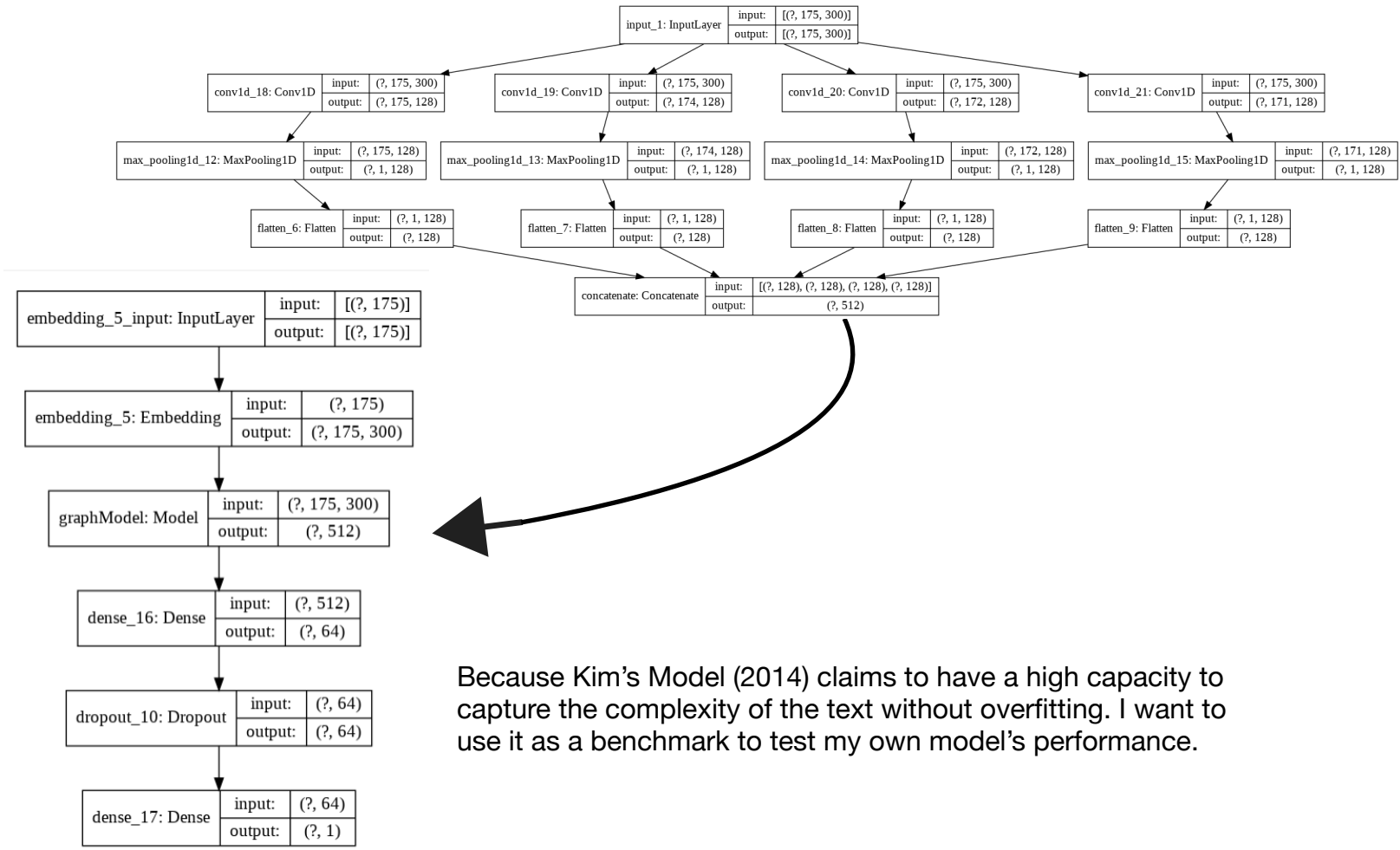
current = current confidence level
oldmin = minimum of current confidence = 0
oldmax = maximum of current confidence = 22.29
newmax = maximum of new confidence = 1
newmin = minimum of new confidence = 0

Through above linear transformation, I rescaled the confidence level to a float number ranged between “0” and “1”. “1” represent 100% sure with the sentiment label, while “0” means the label is irrelevant with the sentiment.

For the training and test sets, I will only use data that has non-zero confidence levels. As for the data which has a 0 as confidence level, I set them as “undetermined data” (approximately 10 % of total data) for models to predict.

Neural Network Architectures

1. Kim’s Model

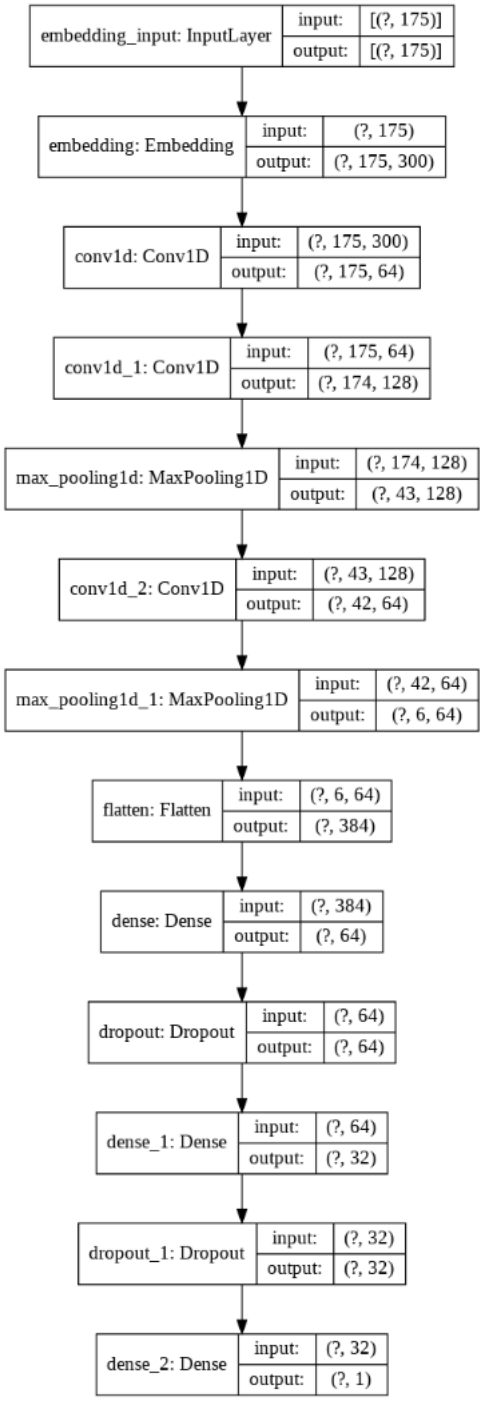


Because Kim’s Model (2014) claims to have a high capacity to capture the complexity of the text without overfitting. I want to use it as a benchmark to test my own model’s performance.

Parameters & Justification of Kim's Hidden Layer

Hidden Layers	Parameters	Reason
Parallel Convolution Layers (enclosed in "graph model")	Kernel size_1: 1 Kernel size_2: 2 Kernel size_3: 4 Kernel size_3: 5	• Since the text data's complexity is relatively simple and expressed in fragmented sentences. The key feature expressing the sentiment lays in short words and short phrases. Therefore, I want to searching features to indicate the sentiment from a few words and short phrases. For example: • 1 word: "good", "disappointed" • 2 words: "not bad", "not good" • 4 words: "I really love it" • 5 words: "I am really disappointed in"
	Filter size: 128 (for each kernel size above)	• At the convolution layers, I want to generate a relatively large set of potential features. So I am looking for 128 sequences of 1, 2, 4 or 5 words that are useful.
	Stride size: 1 (for each kernel size above)	• Since the average comment size is approximately 28 words which is really short. I choose a stride size of 1 to carefully slide through embeddings to capture details instead of a large slide to skip them.
Parallel Maxpooling Layers (enclosed in "graph model")	Pool sizes: 175 (maximum sequence) - kernel_size + 1	• In each features set in the kernel size (1, 2, 4, 5) captured in convolution layers, I want to select the most effective feature set to indict the sentiment of the text in each kernel sizes (1, 2, 4, 5).
Parallel Flatten Layers (enclosed in "graph model")	N/A	• Flatten them so the best four features set can be concatenate
Concatenate (enclosed in "graph model")	N/A	• Enclose the parallel structure
Dense Layer	Filter size: 64	• Change the dimension of vector
Dropout	Dropout Rate: 50%	• Avoiding overfit, "Herd the key features". 50% is an aggressive portion which is desired.

2. Stacked Sequential Model
(2 blocks of convolution layers)



This stacked sequential model is inspired by the VGG 16's hierarchical structure.

From the word distribution plot, the text data only has 26 words on average, and most of them concentrated between 20 words to 60 words. Therefore, this low text complexity only requires one or two blocks of convolution layers to find key features.

Different from Kim's Model, which simultaneously searches for the best features in different word sizes at one single convolution layer, this model looks for different possible combinations of the keywords through multiple convolution layers and select the best of them as crucial feature maps. It creates the diversity in the feature maps for selection, and this is a key advantage that Kim's Model doesn't have.

please see detailed explanations of parameters in the table next page.

Parameters & Justification of Stacked Sequential Hidden Layers

Hidden Layers	Parameters	Reason
1st convolution layer	Kernel size = 1	• Due to the short length of the text data, I want to find single words that are closely related to expressing the sentiment/feeing/attitude in the first convolution layer. For example: “is”, “good”, “am”, “disappointed”.
	Filter size = 64 stride = 1	• To generate a medium size, detailed feature map of single word features for later layers to combine and select from.
2nd convolution layer	Kernel size = 2	• Create features generated from combination of single word in size 2, such as “is | good”, “am | disappointed” etc.
	Filter size = 128 stride = 1	• Different from single words, 2-word feature maps have more diversities, so I preserved more maps (128). We can skip no details given the short comment sizes on average, so stride 1 is reasonable
1st maxpooling layer	Pooling size = 4	• Downsampling, from every set of 4 two-word feature, choose the best one
3rd convolution layer	Kernel size = 2	• Generate combination of combination of 2 words (so in total 4-word feature map) • hopefully find good features such as • “I waited | too long” etc.
	Filter size = 64 Stride = 1	• Shrink map sizes to 64: as complexity of combination increases, more combinations may become meaningless/contradict to each other. • Stride 1: same reason as before, skip no details
2nd maxpooling layer	Pooling size = 7	• Downsampling, from every set of seven 4-word-combination feature, choose the best one.
Flatten Layers	N/A	• Flatten them, transform matrix to vector
Dense Layer	Filter size = 64	• Shrink the vector size to 64
Dropout Layer	Rate = 0.5	• Avoiding overfit, “Herd the key features”.
Dense Layer	Filter size = 32	• Shrink the vector size to 32

Hidden Layers	Parameters	Reason
Dropout Layer	Rate = 0.5	• Avoiding overfit, “Herd the key features” • Added second dropout as an experiment, and the result seems improved. So I kept it.

Other Layers & Parameters

Both architectures uses following input layer, output layer, and complier parameters:

Input Layer	Parameters	Reason
Input Layer	Input size: A matrix of sequences in following dimension: **(# of comments) x (175)**	175 : maximum padding size

Embedding Layer	Parameters	Reason
Embedding Layer	Input size: 175 Embedding matrix size: (# of unique word in comment + 1) x 300 Weights: FastText/GloVe Trainable: False	175: maximum padding size 300: vector size to represent 1 word 1: unknown word Weights: generated from FastText and GloVe Fix the weights since we only use pre-trained embedding

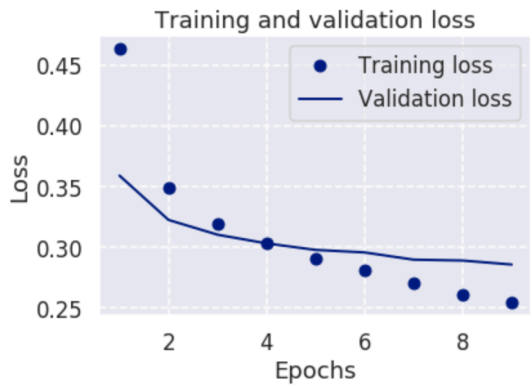
Output Layer	Parameters	Reason
Dense Layer (output)	filter size: 1 activation = “sigmoid”	• For Binary classification output

Validation Size	Batch Size	Optimizer	Loss	Metrics
0.33 • Traditional and reasonable validation size	600 • Theoretically, it should be as large as possible, but I choose 600 to create some “diversity” in data as instructed in the Lab 8	“Adam” • modified learning rate: 0.0001 to avoid overfitting • Best for optimizing binary cross entropy error	“Binary cross entropy” • Binary Classification Problem	“Accuracy”

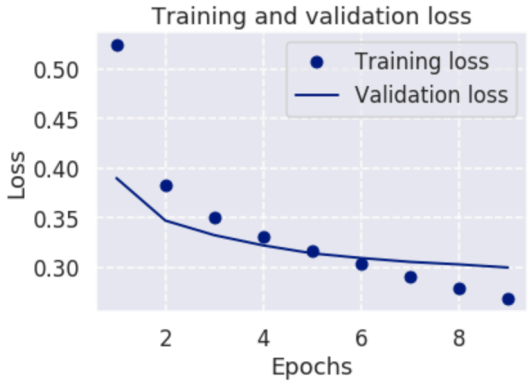
Part d) Performance Analysis

Proof of no overfitting

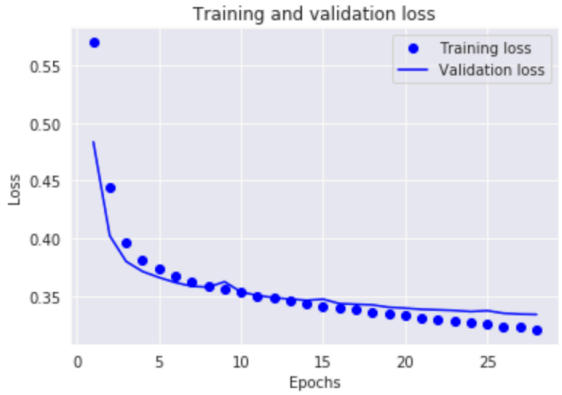
FastText & Kim's



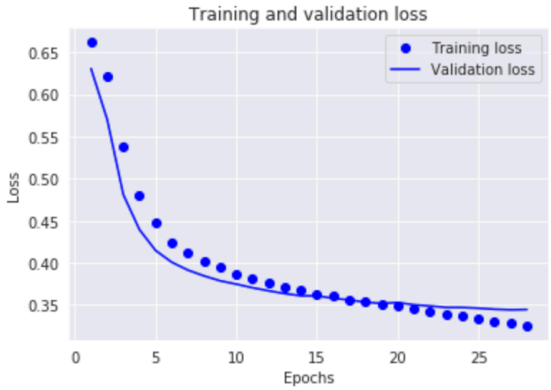
GloVe & Kim's



FastText & Sequential



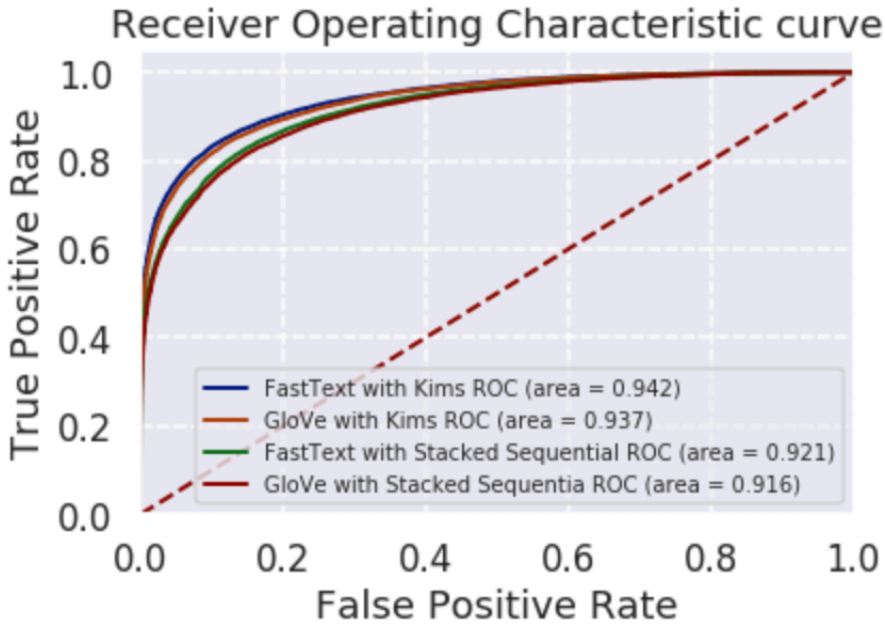
GloVe & Sequential



Performance Evaluation (on Test Set)

Embedding & Architecture	AUROC	Accuracy (cutoff = 0.5)	Confusion Matrix (cutoff = 0.5)
GloVe & Kim's model	0.9371	0.8629	<table><tr><td></td><td>True positive</td><td>True Negative</td></tr><tr><td>Predicted Positive</td><td>36040</td><td>3942</td></tr><tr><td>Predicted Negative</td><td>4270</td><td>15656</td></tr></table>

Embedding & Architecture	AUROC	Accuracy (cutoff = 0.5)	Confusion Matrix (cutoff = 0.5)
GloVe & Stacked Sequential Model	0.9155	0.8425	<table><tr><td></td><td>True positive</td><td>True Negative</td></tr><tr><td>Predicted Positive</td><td>35714</td><td>4840</td></tr><tr><td>Predicted Negative</td><td>4596</td><td>14758</td></tr></table>
FastText & Kim's Model	0.9422	0.8832	<table><tr><td></td><td>True positive</td><td>True Negative</td></tr><tr><td>Predicted Positive</td><td>36652</td><td>4092</td></tr><tr><td>Predicted Negative</td><td>3658</td><td>15506</td></tr></table>
FastText & Stacked Sequential Model	0.9207	0.8485	<table><tr><td></td><td>True positive</td><td>True Negative</td></tr><tr><td>Predicted Positive</td><td>36151</td><td>4919</td></tr><tr><td>Predicted Negative</td><td>4159</td><td>14679</td></tr></table>



From the above evaluation metrics and the plot of ROC curves, we can see that Kim's model generally outperformed my stacked sequential model with both of the embeddings. Having the model architecture fixed, FastText embeddings with subword

techniques generally outperform the GloVe embeddings without subword. So the subword does slightly improve the model's performance given this data set. In conclusion, among the four neural networks, the best architecture is Kim's model, and the best embedding is FastText with the subword feature (trained through Common Crawl).

Reference

1. Kim, Y. (2014, September 3). Convolutional Neural Networks for Sentence Classification. Retrieved December 1, 2019, from https://arxiv.org/abs/1408.5882.

2. Shah, N. (2018, November 15). Open source Emoticons and Emoji detection library: emot. Retrieved December 1, 2019, from https://www.kaggle.com/sudalairajkumar/getting-started-with-text-preprocessing#Conversion-of-Emoji-to-Words.

Appendix

Please see this link to access the appendix
https://colab.research.google.com/drive/1FoA8W83fKYAM4lsVrqWpyn5Vjwz6-wwR

Data Processing for embedding

```
## General Importing
import string
import numpy as np
import re
import pandas as pd
import sklearn.feature_extraction as skprep
from sklearn.metrics import roc_curve, auc
from itertools import compress
import matplotlib.pyplot as plt
import seaborn as sns
import random

random.seed(251121253)
%matplotlib inline


# Keras imports
import tensorflow as tf
import tensorflow.keras as keras
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Input, Embedding, Reshape, MaxPooling1D
from tensorflow.keras.layers import Activation
from tensorflow.keras.layers import Flatten, Dense, Dropout, Lambda
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.optimizers import SGD, RMSprop, Adam
from tensorflow.keras.metrics import categorical_crossentropy, categorical_accuracy
from tensorflow.keras.layers import *
from tensorflow.keras.preprocessing import image, sequence
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

```
The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the
%tensorflow_version 1.x magic: more info.
```

Download the Data to be analyzed

```
!gdown  https://drive.google.com/uc?id=1XRQ-cAOpxAxUFEWcG3fA7twSg8Fmhx6l
```

```
Downloading...
From: https://drive.google.com/uc?id=1XRQ-cAOpxAxUFEWcG3fA7twSg8Fmhx6l
To: /content/Full_Data (1).zip
13.2MB [00:00, 116MB/s]
```

```
!unzip '/content/Full_Data (1).zip'
```

```
Archive:  /content/Full_Data (1).zip
  inflating: Full_Data.csv
```

```
Text_data = pd.read_csv('Full_Data.csv')
```

Download Glove

```
!wget http://nlp.stanford.edu/data/glove.42B.300d.zip
```

```
--2019-12-16 16:58:45--  http://nlp.stanford.edu/data/glove.42B.300d.zip
Resolving nlp.stanford.edu (nlp.stanford.edu)... 171.64.67.140
Connecting to nlp.stanford.edu (nlp.stanford.edu)|171.64.67.140|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://nlp.stanford.edu/data/glove.42B.300d.zip [following]
--2019-12-16 16:58:45--  https://nlp.stanford.edu/data/glove.42B.300d.zip
Connecting to nlp.stanford.edu (nlp.stanford.edu)|171.64.67.140|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://downloads.cs.stanford.edu/nlp/data/glove.42B.300d.zip [following]
--2019-12-16 16:58:45--  http://downloads.cs.stanford.edu/nlp/data/glove.42B.300d.zip
Resolving downloads.cs.stanford.edu (downloads.cs.stanford.edu)... 171.64.64.22
Connecting to downloads.cs.stanford.edu (downloads.cs.stanford.edu)|171.64.64.22|:80... co
HTTP request sent, awaiting response... 200 OK
Length: 1877800501 (1.7G) [application/zip]
Saving to: ‘glove.42B.300d.zip’

glove.42B.300d.zip  100%[===================>]   1.75G  2.09MB/s    in 14m 31s

2019-12-16 17:13:17 (2.06 MB/s) - ‘glove.42B.300d.zip’ saved [1877800501/1877800501]
```

Download Fasttext

```
!wget https://github.com/facebookresearch/fastText/archive/v0.9.1.zip
!unzip v0.9.1.zip
```

```
--2019-12-16 17:13:17--  https://github.com/facebookresearch/fastText/archive/v0.9.1.zip
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/facebookresearch/fastText/zip/v0.9.1 [following]
--2019-12-16 17:13:17--  https://codeload.github.com/facebookresearch/fastText/zip/v0.9.1
Resolving codeload.github.com (codeload.github.com)... 140.82.114.10
Connecting to codeload.github.com (codeload.github.com)|140.82.114.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4327207 (4.1M) [application/zip]
Saving to: ‘v0.9.1.zip’

v0.9.1.zip          100%[===================>]   4.13M  19.9MB/s    in 0.2s

2019-12-16 17:13:18 (19.9 MB/s) - ‘v0.9.1.zip’ saved [4327207/4327207]

Archive:  v0.9.1.zip
b5b7d307274ce00ef52198fbc692ed3bd11d9856
   creating: fastText-0.9.1/
   creating: fastText-0.9.1/.circleci/
  inflating: fastText-0.9.1/.circleci/cmake_test.sh
  inflating: fastText-0.9.1/.circleci/config.yml
  inflating: fastText-0.9.1/.circleci/gcc_test.sh
  inflating: fastText-0.9.1/.circleci/pip_test.sh
  inflating: fastText-0.9.1/.circleci/pull_data.sh
  inflating: fastText-0.9.1/.circleci/python_test.sh
  inflating: fastText-0.9.1/.circleci/run_locally.sh
  inflating: fastText-0.9.1/.circleci/setup_circleimg.sh
  inflating: fastText-0.9.1/.circleci/setup_debian.sh
  inflating: fastText-0.9.1/.gitignore
  inflating: fastText-0.9.1/CMakeLists.txt
  inflating: fastText-0.9.1/CODE_OF_CONDUCT.md
  inflating: fastText-0.9.1/CONTRIBUTING.md
  inflating: fastText-0.9.1/LICENSE
  inflating: fastText-0.9.1/MANIFEST.in
  inflating: fastText-0.9.1/Makefile
  inflating: fastText-0.9.1/README.md
   creating: fastText-0.9.1/alignment/
  inflating: fastText-0.9.1/alignment/README.md
  inflating: fastText-0.9.1/alignment/align.py
  inflating: fastText-0.9.1/alignment/eval.py
  inflating: fastText-0.9.1/alignment/example.sh
  inflating: fastText-0.9.1/alignment/unsup_align.py
  inflating: fastText-0.9.1/alignment/utils.py
  inflating: fastText-0.9.1/classification-example.sh
  inflating: fastText-0.9.1/classification-results.sh
   creating: fastText-0.9.1/crawl/
  inflating: fastText-0.9.1/crawl/README.md
  inflating: fastText-0.9.1/crawl/dedup.cc
  inflating: fastText-0.9.1/crawl/download_crawl.sh
  inflating: fastText-0.9.1/crawl/filter_dedup.sh
  inflating: fastText-0.9.1/crawl/filter_utf8.cc
  inflating: fastText-0.9.1/crawl/process_wet_file.sh
   creating: fastText-0.9.1/docs/
  inflating: fastText-0.9.1/docs/aligned-vectors.md
  inflating: fastText-0.9.1/docs/api.md
  inflating: fastText-0.9.1/docs/cheatsheet.md
  inflating: fastText-0.9.1/docs/crawl-vectors.md
  inflating: fastText-0.9.1/docs/dataset.md
  inflating: fastText-0.9.1/docs/english-vectors.md
  inflating: fastText-0.9.1/docs/faqs.md
  inflating: fastText-0.9.1/docs/language-identification.md
  inflating: fastText-0.9.1/docs/options.md
  inflating: fastText-0.9.1/docs/pretrained-vectors.md
```

```
  inflating: fastText-0.9.1/docs/python-module.md
  inflating: fastText-0.9.1/docs/references.md
  inflating: fastText-0.9.1/docs/supervised-models.md
  inflating: fastText-0.9.1/docs/supervised-tutorial.md
  inflating: fastText-0.9.1/docs/support.md
  inflating: fastText-0.9.1/docs/unsupervised-tutorials.md
  inflating: fastText-0.9.1/eval.py
  inflating: fastText-0.9.1/get-wikimedia.sh
   creating: fastText-0.9.1/python/
  inflating: fastText-0.9.1/python/README.md
  inflating: fastText-0.9.1/python/README.rst
   creating: fastText-0.9.1/python/benchmarks/
  inflating: fastText-0.9.1/python/benchmarks/README.rst
  inflating: fastText-0.9.1/python/benchmarks/get_word_vector.py
   creating: fastText-0.9.1/python/doc/
   creating: fastText-0.9.1/python/doc/examples/
  inflating: fastText-0.9.1/python/doc/examples/FastTextEmbeddingBag.py
  inflating: fastText-0.9.1/python/doc/examples/bin_to_vec.py
  inflating: fastText-0.9.1/python/doc/examples/compute_accuracy.py
  inflating: fastText-0.9.1/python/doc/examples/get_vocab.py
  inflating: fastText-0.9.1/python/doc/examples/train_supervised.py
  inflating: fastText-0.9.1/python/doc/examples/train_unsupervised.py
   creating: fastText-0.9.1/python/fasttext_module/
   creating: fastText-0.9.1/python/fasttext_module/fasttext/
  inflating: fastText-0.9.1/python/fasttext_module/fasttext/FastText.py
  inflating: fastText-0.9.1/python/fasttext_module/fasttext/__init__.py
   creating: fastText-0.9.1/python/fasttext_module/fasttext/pybind/
  inflating: fastText-0.9.1/python/fasttext_module/fasttext/pybind/fasttext_pybind.cc
   creating: fastText-0.9.1/python/fasttext_module/fasttext/tests/
  inflating: fastText-0.9.1/python/fasttext_module/fasttext/tests/__init__.py
  inflating: fastText-0.9.1/python/fasttext_module/fasttext/tests/test_configurations.py
  inflating: fastText-0.9.1/python/fasttext_module/fasttext/tests/test_script.py
   creating: fastText-0.9.1/python/fasttext_module/fasttext/util/
  inflating: fastText-0.9.1/python/fasttext_module/fasttext/util/__init__.py
  inflating: fastText-0.9.1/python/fasttext_module/fasttext/util/util.py
  inflating: fastText-0.9.1/quantization-example.sh
  inflating: fastText-0.9.1/runtests.py
   creating: fastText-0.9.1/scripts/
   creating: fastText-0.9.1/scripts/kbcompletion/
  inflating: fastText-0.9.1/scripts/kbcompletion/README.md
  inflating: fastText-0.9.1/scripts/kbcompletion/data.sh
  inflating: fastText-0.9.1/scripts/kbcompletion/eval.cpp
  inflating: fastText-0.9.1/scripts/kbcompletion/fb15k.sh
  inflating: fastText-0.9.1/scripts/kbcompletion/fb15k237.sh
  inflating: fastText-0.9.1/scripts/kbcompletion/svo.sh
  inflating: fastText-0.9.1/scripts/kbcompletion/wn18.sh
   creating: fastText-0.9.1/scripts/quantization/
  inflating: fastText-0.9.1/scripts/quantization/quantization-results.sh
 extracting: fastText-0.9.1/setup.cfg
  inflating: fastText-0.9.1/setup.py
   creating: fastText-0.9.1/src/
  inflating: fastText-0.9.1/src/args.cc
  inflating: fastText-0.9.1/src/args.h
  inflating: fastText-0.9.1/src/densematrix.cc
  inflating: fastText-0.9.1/src/densematrix.h
  inflating: fastText-0.9.1/src/dictionary.cc
  inflating: fastText-0.9.1/src/dictionary.h
  inflating: fastText-0.9.1/src/fasttext.cc
  inflating: fastText-0.9.1/src/fasttext.h
  inflating: fastText-0.9.1/src/loss.cc
  inflating: fastText-0.9.1/src/loss.h
  inflating: fastText-0.9.1/src/main.cc
  inflating: fastText-0.9.1/src/matrix.cc
  inflating: fastText-0.9.1/src/matrix.h
```

```
  inflating: fastText-0.9.1/src/meter.cc
  inflating: fastText-0.9.1/src/meter.h
  inflating: fastText-0.9.1/src/model.cc
  inflating: fastText-0.9.1/src/model.h
  inflating: fastText-0.9.1/src/productquantizer.cc
  inflating: fastText-0.9.1/src/productquantizer.h
  inflating: fastText-0.9.1/src/quantmatrix.cc
  inflating: fastText-0.9.1/src/quantmatrix.h
  inflating: fastText-0.9.1/src/real.h
  inflating: fastText-0.9.1/src/utils.cc
  inflating: fastText-0.9.1/src/utils.h
  inflating: fastText-0.9.1/src/vector.cc
  inflating: fastText-0.9.1/src/vector.h
   creating: fastText-0.9.1/tests/
  inflating: fastText-0.9.1/tests/fetch_test_data.sh
   creating: fastText-0.9.1/website/
  inflating: fastText-0.9.1/website/README.md
   creating: fastText-0.9.1/website/blog/
  inflating: fastText-0.9.1/website/blog/2016-08-18-blog-post.md
  inflating: fastText-0.9.1/website/blog/2017-05-02-blog-post.md
  inflating: fastText-0.9.1/website/blog/2017-10-02-blog-post.md
  inflating: fastText-0.9.1/website/blog/2019-06-25-blog-post.md
   creating: fastText-0.9.1/website/core/
  inflating: fastText-0.9.1/website/core/Footer.js
  inflating: fastText-0.9.1/website/package.json
   creating: fastText-0.9.1/website/pages/
   creating: fastText-0.9.1/website/pages/en/
  inflating: fastText-0.9.1/website/pages/en/index.js
  inflating: fastText-0.9.1/website/sidebars.json
  inflating: fastText-0.9.1/website/siteConfig.js
   creating: fastText-0.9.1/website/static/
   creating: fastText-0.9.1/website/static/docs/
   creating: fastText-0.9.1/website/static/docs/en/
   creating: fastText-0.9.1/website/static/docs/en/html/
 extracting: fastText-0.9.1/website/static/docs/en/html/.classfasttext_1_1QMatrix-members.
  inflating: fastText-0.9.1/website/static/docs/en/html/annotated.html
  inflating: fastText-0.9.1/website/static/docs/en/html/annotated_dup.js
  inflating: fastText-0.9.1/website/static/docs/en/html/args_8cc.html
  inflating: fastText-0.9.1/website/static/docs/en/html/args_8h.html
  inflating: fastText-0.9.1/website/static/docs/en/html/args_8h.js
  inflating: fastText-0.9.1/website/static/docs/en/html/args_8h_source.html
 extracting: fastText-0.9.1/website/static/docs/en/html/bc_s.png
  inflating: fastText-0.9.1/website/static/docs/en/html/bdwn.png
  inflating: fastText-0.9.1/website/static/docs/en/html/classes.html
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Args-members.html
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Args.html
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Args.js
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Dictionary-member
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Dictionary.html
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Dictionary.js
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1FastText-members.
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1FastText.html
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1FastText.js
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Matrix-members.ht
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Matrix.html
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Matrix.js
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Model-members.htm
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Model.html
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Model.js
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1ProductQuantizer-
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1ProductQuantizer.
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1ProductQuantizer.
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1QMatrix-members.h
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1QMatrix.html
```

```
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1QMatrix.js
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Vector-members.ht
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Vector.html
  inflating: fastText-0.9.1/website/static/docs/en/html/classfasttext_1_1Vector.js
  inflating: fastText-0.9.1/website/static/docs/en/html/closed.png
  inflating: fastText-0.9.1/website/static/docs/en/html/dictionary_8cc.html
  inflating: fastText-0.9.1/website/static/docs/en/html/dictionary_8h.html
  inflating: fastText-0.9.1/website/static/docs/en/html/dictionary_8h.js
  inflating: fastText-0.9.1/website/static/docs/en/html/dictionary_8h_source.html
  inflating: fastText-0.9.1/website/static/docs/en/html/dir_68267d1309a1af8e8297ef4c3efbcd
  inflating: fastText-0.9.1/website/static/docs/en/html/dir_68267d1309a1af8e8297ef4c3efbcd
 extracting: fastText-0.9.1/website/static/docs/en/html/doc.png
  inflating: fastText-0.9.1/website/static/docs/en/html/doxygen.css
 extracting: fastText-0.9.1/website/static/docs/en/html/doxygen.png
  inflating: fastText-0.9.1/website/static/docs/en/html/dynsections.js
  inflating: fastText-0.9.1/website/static/docs/en/html/fasttext_8cc.html
  inflating: fastText-0.9.1/website/static/docs/en/html/fasttext_8h.html
  inflating: fastText-0.9.1/website/static/docs/en/html/fasttext_8h.js
  inflating: fastText-0.9.1/website/static/docs/en/html/fasttext_8h_source.html
  inflating: fastText-0.9.1/website/static/docs/en/html/favicon.png
  inflating: fastText-0.9.1/website/static/docs/en/html/files.html
  inflating: fastText-0.9.1/website/static/docs/en/html/files.js
 extracting: fastText-0.9.1/website/static/docs/en/html/folderclosed.png
 extracting: fastText-0.9.1/website/static/docs/en/html/folderopen.png
  inflating: fastText-0.9.1/website/static/docs/en/html/functions.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_0x7e.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_b.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_c.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_d.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_dup.js
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_e.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_f.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_func.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_g.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_h.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_i.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_k.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_l.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_m.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_n.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_o.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_p.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_q.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_r.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_s.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_t.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_u.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_v.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_vars.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_w.html
  inflating: fastText-0.9.1/website/static/docs/en/html/functions_z.html
  inflating: fastText-0.9.1/website/static/docs/en/html/globals.html
  inflating: fastText-0.9.1/website/static/docs/en/html/globals_defs.html
  inflating: fastText-0.9.1/website/static/docs/en/html/globals_func.html
  inflating: fastText-0.9.1/website/static/docs/en/html/index.html
  inflating: fastText-0.9.1/website/static/docs/en/html/jquery.js
  inflating: fastText-0.9.1/website/static/docs/en/html/main_8cc.html
  inflating: fastText-0.9.1/website/static/docs/en/html/main_8cc.js
  inflating: fastText-0.9.1/website/static/docs/en/html/matrix_8cc.html
  inflating: fastText-0.9.1/website/static/docs/en/html/matrix_8h.html
  inflating: fastText-0.9.1/website/static/docs/en/html/matrix_8h_source.html
  inflating: fastText-0.9.1/website/static/docs/en/html/menu.js
  inflating: fastText-0.9.1/website/static/docs/en/html/menudata.js
  inflating: fastText-0.9.1/website/static/docs/en/html/model_8cc.html
```

```
  inflating: fastText-0.9.1/website/static/docs/en/html/model_8h.html
  inflating: fastText-0.9.1/website/static/docs/en/html/model_8h.js
  inflating: fastText-0.9.1/website/static/docs/en/html/model_8h_source.html
  inflating: fastText-0.9.1/website/static/docs/en/html/namespacefasttext.html
  inflating: fastText-0.9.1/website/static/docs/en/html/namespacefasttext.js
  inflating: fastText-0.9.1/website/static/docs/en/html/namespacefasttext_1_1utils.html
  inflating: fastText-0.9.1/website/static/docs/en/html/namespacemembers.html
  inflating: fastText-0.9.1/website/static/docs/en/html/namespacemembers_enum.html
  inflating: fastText-0.9.1/website/static/docs/en/html/namespacemembers_func.html
  inflating: fastText-0.9.1/website/static/docs/en/html/namespacemembers_type.html
  inflating: fastText-0.9.1/website/static/docs/en/html/namespaces.html
  inflating: fastText-0.9.1/website/static/docs/en/html/namespaces.js
 extracting: fastText-0.9.1/website/static/docs/en/html/nav_f.png
  inflating: fastText-0.9.1/website/static/docs/en/html/nav_g.png
  inflating: fastText-0.9.1/website/static/docs/en/html/nav_h.png
  inflating: fastText-0.9.1/website/static/docs/en/html/navtree.css
  inflating: fastText-0.9.1/website/static/docs/en/html/navtree.js
  inflating: fastText-0.9.1/website/static/docs/en/html/navtreedata.js
  inflating: fastText-0.9.1/website/static/docs/en/html/navtreeindex0.js
  inflating: fastText-0.9.1/website/static/docs/en/html/navtreeindex1.js
  inflating: fastText-0.9.1/website/static/docs/en/html/open.png
  inflating: fastText-0.9.1/website/static/docs/en/html/productquantizer_8cc.html
  inflating: fastText-0.9.1/website/static/docs/en/html/productquantizer_8cc.js
  inflating: fastText-0.9.1/website/static/docs/en/html/productquantizer_8h.html
  inflating: fastText-0.9.1/website/static/docs/en/html/productquantizer_8h_source.html
  inflating: fastText-0.9.1/website/static/docs/en/html/qmatrix_8cc.html
  inflating: fastText-0.9.1/website/static/docs/en/html/qmatrix_8h.html
  inflating: fastText-0.9.1/website/static/docs/en/html/qmatrix_8h_source.html
  inflating: fastText-0.9.1/website/static/docs/en/html/real_8h.html
  inflating: fastText-0.9.1/website/static/docs/en/html/real_8h.js
  inflating: fastText-0.9.1/website/static/docs/en/html/real_8h_source.html
  inflating: fastText-0.9.1/website/static/docs/en/html/resize.js
   creating: fastText-0.9.1/website/static/docs/en/html/search/
 extracting: fastText-0.9.1/website/static/docs/en/html/search/.files_7.html.StRRNc
 extracting: fastText-0.9.1/website/static/docs/en/html/search/.variables_a.html.1MGQ27
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_0.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_0.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_1.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_1.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_10.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_10.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_11.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_11.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_12.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_12.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_13.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_13.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_14.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_14.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_15.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_15.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_16.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_16.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_17.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_17.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_2.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_2.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_3.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_3.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_4.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_4.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_5.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_5.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_6.html
```

```
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_6.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_7.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_7.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_8.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_8.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_9.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_9.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_a.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_a.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_b.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_b.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_c.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_c.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_d.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_d.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_e.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_e.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_f.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/all_f.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_0.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_0.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_1.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_1.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_2.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_2.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_3.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_3.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_4.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_4.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_5.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_5.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_6.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_6.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_7.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_7.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_8.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/classes_8.js
 extracting: fastText-0.9.1/website/static/docs/en/html/search/close.png
  inflating: fastText-0.9.1/website/static/docs/en/html/search/defines_0.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/defines_0.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/defines_1.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/defines_1.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/defines_2.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/defines_2.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/defines_3.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/defines_3.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enums_0.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enums_0.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enums_1.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enums_1.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enums_2.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enums_2.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_0.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_0.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_1.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_1.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_2.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_2.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_3.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_3.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_4.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_4.js
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_5.html
  inflating: fastText-0.9.1/website/static/docs/en/html/search/enumvalues_5.js
```

```
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_0.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_0.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_1.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_1.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_2.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_2.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_3.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_3.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_4.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_4.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_5.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_5.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_6.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_6.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_7.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_7.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_8.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/files_8.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_0.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_0.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_1.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_1.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_10.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_10.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_11.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_11.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_12.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_12.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_13.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_13.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_14.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_14.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_15.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_15.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_16.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_16.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_17.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_17.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_2.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_2.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_3.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_3.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_4.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_4.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_5.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_5.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_6.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_6.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_7.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_7.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_8.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_8.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_9.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_9.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_a.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_a.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_b.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_b.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_c.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_c.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_d.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_d.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_e.html
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_e.js
inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_f.html
```

```
 inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_f.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/functions_f.js
extracting: fastText-0.9.1/website/static/docs/en/html/search/mag_sel.png
 inflating: fastText-0.9.1/website/static/docs/en/html/search/namespaces_0.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/namespaces_0.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/nomatches.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/search.css
 inflating: fastText-0.9.1/website/static/docs/en/html/search/search.js
extracting: fastText-0.9.1/website/static/docs/en/html/search/search_l.png
 inflating: fastText-0.9.1/website/static/docs/en/html/search/search_m.png
extracting: fastText-0.9.1/website/static/docs/en/html/search/search_r.png
 inflating: fastText-0.9.1/website/static/docs/en/html/search/searchdata.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/typedefs_0.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/typedefs_0.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/typedefs_1.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/typedefs_1.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_0.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_0.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_1.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_1.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_10.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_10.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_11.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_11.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_12.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_12.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_13.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_13.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_2.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_2.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_3.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_3.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_4.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_4.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_5.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_5.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_6.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_6.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_7.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_7.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_8.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_8.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_9.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_9.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_a.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_a.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_b.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_b.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_c.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_c.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_d.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_d.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_e.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_e.js
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_f.html
 inflating: fastText-0.9.1/website/static/docs/en/html/search/variables_f.js
 inflating: fastText-0.9.1/website/static/docs/en/html/splitbar.png
 inflating: fastText-0.9.1/website/static/docs/en/html/structfasttext_1_1Node-members.htm
 inflating: fastText-0.9.1/website/static/docs/en/html/structfasttext_1_1Node.html
 inflating: fastText-0.9.1/website/static/docs/en/html/structfasttext_1_1Node.js
 inflating: fastText-0.9.1/website/static/docs/en/html/structfasttext_1_1entry-members.ht
 inflating: fastText-0.9.1/website/static/docs/en/html/structfasttext_1_1entry.html
 inflating: fastText-0.9.1/website/static/docs/en/html/structfasttext_1_1entry.js
extracting: fastText-0.9.1/website/static/docs/en/html/sync_off.png
extracting: fastText-0.9.1/website/static/docs/en/html/sync_on.png
```

```
extracting: fastText-0.9.1/website/static/docs/en/html/sync_on.png
extracting: fastText-0.9.1/website/static/docs/en/html/tab_a.png
extracting: fastText-0.9.1/website/static/docs/en/html/tab_b.png
extracting: fastText-0.9.1/website/static/docs/en/html/tab_h.png
extracting: fastText-0.9.1/website/static/docs/en/html/tab_s.png
 inflating: fastText-0.9.1/website/static/docs/en/html/tabs.css
 inflating: fastText-0.9.1/website/static/docs/en/html/utils_8cc.html
 inflating: fastText-0.9.1/website/static/docs/en/html/utils_8cc.js
 inflating: fastText-0.9.1/website/static/docs/en/html/utils_8h.html
 inflating: fastText-0.9.1/website/static/docs/en/html/utils_8h.js
 inflating: fastText-0.9.1/website/static/docs/en/html/utils_8h_source.html
 inflating: fastText-0.9.1/website/static/docs/en/html/vector_8cc.html
 inflating: fastText-0.9.1/website/static/docs/en/html/vector_8cc.js
 inflating: fastText-0.9.1/website/static/docs/en/html/vector_8h.html
 inflating: fastText-0.9.1/website/static/docs/en/html/vector_8h.js
 inflating: fastText-0.9.1/website/static/docs/en/html/vector_8h_source.html
 inflating: fastText-0.9.1/website/static/fasttext.css
  creating: fastText-0.9.1/website/static/img/
  creating: fastText-0.9.1/website/static/img/authors/
 inflating: fastText-0.9.1/website/static/img/authors/armand_joulin.jpg
 inflating: fastText-0.9.1/website/static/img/authors/christian_puhrsch.png
 inflating: fastText-0.9.1/website/static/img/authors/edouard_grave.jpeg
 inflating: fastText-0.9.1/website/static/img/authors/piotr_bojanowski.jpg
 inflating: fastText-0.9.1/website/static/img/authors/tomas_mikolov.jpg
  creating: fastText-0.9.1/website/static/img/blog/
 inflating: fastText-0.9.1/website/static/img/blog/2016-08-18-blog-post-img1.png
 inflating: fastText-0.9.1/website/static/img/blog/2016-08-18-blog-post-img2.png
 inflating: fastText-0.9.1/website/static/img/blog/2017-05-02-blog-post-img1.jpg
 inflating: fastText-0.9.1/website/static/img/blog/2017-05-02-blog-post-img2.jpg
 inflating: fastText-0.9.1/website/static/img/blog/2017-10-02-blog-post-img1.png
 inflating: fastText-0.9.1/website/static/img/cbo_vs_skipgram.png
 inflating: fastText-0.9.1/website/static/img/fasttext-icon-api.png
 inflating: fastText-0.9.1/website/static/img/fasttext-icon-bg-web.png
 inflating: fastText-0.9.1/website/static/img/fasttext-icon-color-square.png
 inflating: fastText-0.9.1/website/static/img/fasttext-icon-color-web.png
 inflating: fastText-0.9.1/website/static/img/fasttext-icon-faq.png
 inflating: fastText-0.9.1/website/static/img/fasttext-icon-tutorial.png
 inflating: fastText-0.9.1/website/static/img/fasttext-icon-white-web.png
 inflating: fastText-0.9.1/website/static/img/fasttext-logo-color-web.png
 inflating: fastText-0.9.1/website/static/img/fasttext-logo-white-web.png
 inflating: fastText-0.9.1/website/static/img/logo-color.png
 inflating: fastText-0.9.1/website/static/img/model-black.png
 inflating: fastText-0.9.1/website/static/img/model-blue.png
 inflating: fastText-0.9.1/website/static/img/model-red.png
 inflating: fastText-0.9.1/website/static/img/ogimage.png
 inflating: fastText-0.9.1/website/static/img/oss_logo.png
 inflating: fastText-0.9.1/wikifil.pl
 inflating: fastText-0.9.1/word-vector-example.sh
```

```
%cd fastText-0.9.1
!make
```

```
/content/fastText-0.9.1
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/args.cc
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/matrix.cc
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/dictionary.cc
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/loss.cc
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/productquantizer.
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/densematrix.cc
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/quantmatrix.cc
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/vector.cc
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/model.cc
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/utils.cc
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/meter.cc
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG -c src/fasttext.cc
src/fasttext.cc: In member function ‘void fasttext::FastText::quantize(const fasttext::Arg
src/fasttext.cc:323:45: warning: ‘std::vector<int> fasttext::FastText::selectEmbeddings(in
     auto idx = selectEmbeddings(qargs.cutoff);
                                             ^
src/fasttext.cc:293:22: note: declared here
 std::vector<int32_t> FastText::selectEmbeddings(int32_t cutoff) const {
                      ^~~~~~~~
src/fasttext.cc: In member function ‘void fasttext::FastText::lazyComputeWordVectors()’:
src/fasttext.cc:551:40: warning: ‘void fasttext::FastText::precomputeWordVectors(fasttext:
     precomputeWordVectors(*wordVectors_);
                                        ^
src/fasttext.cc:534:6: note: declared here
 void FastText::precomputeWordVectors(DenseMatrix& wordVectors) {
      ^~~~~~~~
c++ -pthread -std=c++0x -march=native -O3 -funroll-loops -DNDEBUG args.o matrix.o dictiona
```

```
!ls
```

```
alignment                 fasttext.o              README.md
args.o                    get-wikimedia.sh        runtests.py
classification-example.sh LICENSE                 scripts
classification-results.sh loss.o                  setup.cfg
CMakeLists.txt            Makefile                setup.py
CODE_OF_CONDUCT.md        MANIFEST.in             src
CONTRIBUTING.md           matrix.o                tests
crawl                     meter.o                 utils.o
densematrix.o             model.o                 vector.o
dictionary.o              productquantizer.o      website
docs                      python                  wikifil.pl
eval.py                   quantization-example.sh word-vector-example.sh
fasttext                  quantmatrix.o
```

Processing the data in a way that Fasttext and Glove can use it

This include processing all the puntucation marks etc.

we will clean the text as following:

1. Remove any URLs present in tweets as they are not significant in sentiment analysis.
2. Replace any emojis with the text they represent as emojis or emoticons plays an important role in representing a sentimen
3. Replace contractions with their full forms.
4. Remove mentions as they also do not weigh in sentiment analyzing.
5. Remove punctuations.
6. Remove some of the stopwords that is not going to help on analyzing the sentiment

```
%cd /content
```

```
/content
```

```
!ls
```

```
 fastText-0.9.1       Full_Data.csv        sample_data
'Full_Data (1).zip'   glove.42B.300d.zip   v0.9.1.zip
```

```
!pip install textsearch
```

```
Collecting textsearch
  Downloading https://files.pythonhosted.org/packages/42/a8/03407021f9555043de5492a2bd7a35
Collecting Unidecode
  Downloading https://files.pythonhosted.org/packages/d0/42/d9edfed04228bacea2d824904cae36
     |████████████████████████████████| 245kB 5.0MB/s
Collecting pyahocorasick
  Downloading https://files.pythonhosted.org/packages/f4/9f/f0d8e8850e12829eea2e778f1c90e3
     |████████████████████████████████| 317kB 45.9MB/s
Building wheels for collected packages: pyahocorasick
  Building wheel for pyahocorasick (setup.py) ... done
  Created wheel for pyahocorasick: filename=pyahocorasick-1.4.0-cp36-cp36m-linux_x86_64.wh
  Stored in directory: /root/.cache/pip/wheels/0a/90/61/87a55f5b459792fbb2b7ba6b31721b06ff
Successfully built pyahocorasick
Installing collected packages: Unidecode, pyahocorasick, textsearch
Successfully installed Unidecode-1.1.1 pyahocorasick-1.4.0 textsearch-0.0.17
```

```
!pip install contractions
import contractions
```

```
Collecting contractions
  Downloading https://files.pythonhosted.org/packages/85/41/c3dfd5feb91a8d587ed1a59f553f07
Requirement already satisfied: textsearch in /usr/local/lib/python3.6/dist-packages (from
Requirement already satisfied: pyahocorasick in /usr/local/lib/python3.6/dist-packages (fr
Requirement already satisfied: Unidecode in /usr/local/lib/python3.6/dist-packages (from t
Installing collected packages: contractions
Successfully installed contractions-0.0.24
```

```
##Function to remove URl
def remove_URL(text):
    """Remove URLs from a sample string"""
    return re.sub(r"http\S+", "", text)
```

From the website, we have the following dictionary of Emoicons

```
EMOTICONS = {
    u":-)":"Happy face or smiley",
    u":)":"Happy face or smiley",
    u":-]":"Happy face or smiley",
    u":]":"Happy face or smiley",
    u":-3":"Happy face smiley",
    u":3":"Happy face smiley",
    u":->":"Happy face smiley",
```

```
u":>":"Happy face smiley",
u"8-)":"Happy face smiley",
u":o)":"Happy face smiley",
u":-}":"Happy face smiley",
u":}":"Happy face smiley",
u":-)":"Happy face smiley",
u":c)":"Happy face smiley",
u":^)":"Happy face smiley",
u"=]":"Happy face smiley",
u"=)":"Happy face smiley",
u":-D":"Laughing, big grin or laugh with glasses",
u":D":"Laughing, big grin or laugh with glasses",
u"8-D":"Laughing, big grin or laugh with glasses",
u"8D":"Laughing, big grin or laugh with glasses",
u"X-D":"Laughing, big grin or laugh with glasses",
u"XD":"Laughing, big grin or laugh with glasses",
u"=D":"Laughing, big grin or laugh with glasses",
u"=3":"Laughing, big grin or laugh with glasses",
u"B^D":"Laughing, big grin or laugh with glasses",
u":-))":"Very happy",
u":-(":"Frown, sad, andry or pouting",
u":-(":"Frown, sad, andry or pouting",
u":(":"Frown, sad, andry or pouting",
u":-c":"Frown, sad, andry or pouting",
u":c":"Frown, sad, andry or pouting",
u":-<":"Frown, sad, andry or pouting",
u":<":"Frown, sad, andry or pouting",
u":-[":"Frown, sad, andry or pouting",
u":[":"Frown, sad, andry or pouting",
u":-||":"Frown, sad, andry or pouting",
u">:[":"Frown, sad, andry or pouting",
u":{":"Frown, sad, andry or pouting",
u":@":"Frown, sad, andry or pouting",
u">:(":"Frown, sad, andry or pouting",
u":'-(":"Crying",
u":'(":"Crying",
u":'-)":"Tears of happiness",
u":')":"Tears of happiness",
u"D-':":"Horror",
u"D:<":"Disgust",
u"D:":"Sadness",
u"D8":"Great dismay",
u"D;":"Great dismay",
u"D=":"Great dismay",
u"DX":"Great dismay",
u":-O":"Surprise",
u":O":"Surprise",
u":-o":"Surprise",
u":o":"Surprise",
u":-0":"Shock",
u"8-0":"Yawn",
u">:O":"Yawn",
u":-*":"Kiss",
u":*":"Kiss",
u":X":"Kiss",
u";-)":"Wink or smirk",
u";)":"Wink or smirk",
```

```
    u"*-)":"Wink or smirk",
    u"*)":"Wink or smirk",
    u";-]":"Wink or smirk",
    u";]":"Wink or smirk",
    u";^)":"Wink or smirk",
    u":-,":"Wink or smirk",
    u";D":"Wink or smirk",
    u":-P":"Tongue sticking out, cheeky, playful or blowing a raspberry",
    u":P":"Tongue sticking out, cheeky, playful or blowing a raspberry",
    u"X-P":"Tongue sticking out, cheeky, playful or blowing a raspberry",
    u"XP":"Tongue sticking out, cheeky, playful or blowing a raspberry",
    u":-Þ":"Tongue sticking out, cheeky, playful or blowing a raspberry",
    u":Þ":"Tongue sticking out, cheeky, playful or blowing a raspberry",
    u":b":"Tongue sticking out, cheeky, playful or blowing a raspberry",
    u"d:":"Tongue sticking out, cheeky, playful or blowing a raspberry",
    u"=p":"Tongue sticking out, cheeky, playful or blowing a raspberry",
    u">:P":"Tongue sticking out, cheeky, playful or blowing a raspberry",
    u":-/":"Skeptical, annoyed, undecided, uneasy or hesitant",
    u":/":"Skeptical, annoyed, undecided, uneasy or hesitant",
    u":-[.]":"Skeptical, annoyed, undecided, uneasy or hesitant",
    u">:[(\)]":"Skeptical, annoyed, undecided, uneasy or hesitant",
    u">:/":"Skeptical, annoyed, undecided, uneasy or hesitant",
    u":[(\)]":"Skeptical, annoyed, undecided, uneasy or hesitant",
    u"=/":"Skeptical, annoyed, undecided, uneasy or hesitant",
    u"=[(\)]":"Skeptical, annoyed, undecided, uneasy or hesitant",
    u":L":"Skeptical, annoyed, undecided, uneasy or hesitant",
    u"=L":"Skeptical, annoyed, undecided, uneasy or hesitant",
    u":S":"Skeptical, annoyed, undecided, uneasy or hesitant",
    u":-|":"Straight face",
    u":|":"Straight face",
    u":$":"Embarrassed or blushing",
    u":-x":"Sealed lips or wearing braces or tongue-tied",
    u":x":"Sealed lips or wearing braces or tongue-tied",
    u":-#":"Sealed lips or wearing braces or tongue-tied",
    u":#":"Sealed lips or wearing braces or tongue-tied",
    u":-&":"Sealed lips or wearing braces or tongue-tied",
    u":&":"Sealed lips or wearing braces or tongue-tied",
    u"O:-)":"Angel, saint or innocent",
    u"O:)":"Angel, saint or innocent",
    u"0:-3":"Angel, saint or innocent",
    u"0:3":"Angel, saint or innocent",
    u"0:-)":"Angel, saint or innocent",
    u"0:)":"Angel, saint or innocent",
    u":-b":"Tongue sticking out, cheeky, playful or blowing a raspberry",
    u"0;^)":"Angel, saint or innocent",
    u">:-)":"Evil or devilish",
    u">:)":"Evil or devilish",
    u"}:-)":"Evil or devilish",
    u"}:)":"Evil or devilish",
    u"3:-)":"Evil or devilish",
    u"3:)":"Evil or devilish",
    u">;)":"Evil or devilish",
    u"|;-)":"Cool",
    u"|-O":"Bored",
    u":-J":"Tongue-in-cheek",
    u"#-)":"Party all night",
    u"%-)":"Drunk or confused",
    u"%)":"Drunk or confused"
```

```
u"8) : Drunk or confused",
u":-###..":"Being sick",
u":###..":"Being sick",
u"<:-|":"Dump",
u"(>_<)":"Troubled",
u"(>_<)>":"Troubled",
u"(';')":"Baby",
u"(^^>``":"Nervous or Embarrassed or Troubled or Shy or Sweat drop",
u"(^_^;)":"Nervous or Embarrassed or Troubled or Shy or Sweat drop",
u"(-_-;)":"Nervous or Embarrassed or Troubled or Shy or Sweat drop",
u"(~_~;) (・.・;)":"Nervous or Embarrassed or Troubled or Shy or Sweat drop",
u"(-_-)zzz":"Sleeping",
u"(^_-)":"Wink",
u"((+_+))":"Confused",
u"(+o+)":"Confused",
u"(o|o)":"Ultraman",
u"^_^":"Joyful",
u"(^_^)/":"Joyful",
u"(^O^)／":"Joyful",
u"(^o^)／":"Joyful",
u"(__)":"Kowtow as a sign of respect, or dogeza for apology",
u"_(._.)_":"Kowtow as a sign of respect, or dogeza for apology",
u"<(_ _)>":"Kowtow as a sign of respect, or dogeza for apology",
u"<m(__)m>":"Kowtow as a sign of respect, or dogeza for apology",
u"m(__)m":"Kowtow as a sign of respect, or dogeza for apology",
u"m(_ _)m":"Kowtow as a sign of respect, or dogeza for apology",
u"('_')":"Sad or Crying",
u"(/_;)":"Sad or Crying",
u"(T_T) (;_;)":"Sad or Crying",
u"(;_;":"Sad of Crying",
u"(;_:)":"Sad or Crying",
u"(;O;)":"Sad or Crying",
u"(:_;)":"Sad or Crying",
u"(ToT)":"Sad or Crying",
u";_;":"Sad or Crying",
u";-;":"Sad or Crying",
u";n;":"Sad or Crying",
u";;":"Sad or Crying",
u"Q.Q":"Sad or Crying",
u"T.T":"Sad or Crying",
u"QQ":"Sad or Crying",
u"Q_Q":"Sad or Crying",
u"(-.-)":"Shame",
u"(-_-)":"Shame",
u"(——)":"Shame",
u"(; —_—)":"Shame",
u"(=_=)":"Tired",
u"(=^・^=)":"cat",
u"(=^・・^=)":"cat",
u"=_^= ":"cat",
u"(..)":"Looking down",
u"(._.)":"Looking down",
u"^m^":"Giggling with hand covering mouth",
u"(・・?":"Confusion",
u"(?_?)":"Confusion",
u">^_^<":"Normal Laugh",
u"<^!^>":"Normal Laugh",
u"^/^":"Normal Laugh",
```

```
    u" (*^_^*) " :"Normal Laugh",
    u"(^<^) (^.^)":"Normal Laugh",
    u"(^^)":"Normal Laugh",
    u"(^.^)":"Normal Laugh",
    u"(^_^.)":"Normal Laugh",
    u"(^_^)":"Normal Laugh",
    u"(^^)":"Normal Laugh",
    u"(^J^)":"Normal Laugh",
    u"(*^.^*)":"Normal Laugh",
    u"(^—^) ":"Normal Laugh",
    u"(#^.^#)":"Normal Laugh",
    u" (^—^) ":"Waving",
    u"(;_;)/~~~":"Waving",
    u"(^.^)/~~~":"Waving",
    u"(-_-)/~~~ ($··)/~~~":"Waving",
    u"(T_T)/~~~":"Waving",
    u"(ToT)/~~~":"Waving",
    u"(*^0^*)":"Excited",
    u"(*_*)":"Amazed",
    u"(*_*;":"Amazed",
    u"(+_+) (@_@)":"Amazed",
    u"(*^^)v":"Laughing,Cheerful",
    u"(^_^)v":"Laughing,Cheerful",
    u"((d[-_-]b))":"Headphones,Listening to music",
    u'(-"-)':"Worried",
    u"(——;)":"Worried",
    u"(^0_0^)":"Eyeglasses",
    u"( ^ v ^ )":"Happy",
    u"( ^ u ^ )":"Happy",
    u"(^)o(^)":"Happy",
    u"(^O^)":"Happy",
    u"(^o^)":"Happy",
    u")^o^(":"Happy",
    u":O o_O":"Surprised",
    u"o_0":"Surprised",
    u"o.O":"Surpised",
    u"(o.o)":"Surprised",
    u"oO":"Surprised",
    u"(* ¯ m ¯ )":"Dissatisfied",
    u"('A`)":"Snubbed or Deflated"

}


EMOTICONS2 = {
    u":-\)":"Happy face or smiley",
    u":\)":"Happy face or smiley",
    u":-\]":"Happy face or smiley",
    u":\]":"Happy face or smiley",
    u":-3":"Happy face smiley",
    u":3":"Happy face smiley",
    u":->":"Happy face smiley",
    u":>":"Happy face smiley",
    u"8-\)":"Happy face smiley",
    u":o\)":"Happy face smiley",
    u":-\}":"Happy face smiley",
    u":\}":"Happy face smiley",
    u":-\)":"Happy face smiley",
```

```
u":c\)":"Happy face smiley",
u":\^\)":"Happy face smiley",
u"=\]":"Happy face smiley",
u"=\)":"Happy face smiley",
u":-D":"Laughing, big grin or laugh with glasses",
u":D":"Laughing, big grin or laugh with glasses",
u"8-D":"Laughing, big grin or laugh with glasses",
u"8D":"Laughing, big grin or laugh with glasses",
u"X-D":"Laughing, big grin or laugh with glasses",
u"XD":"Laughing, big grin or laugh with glasses",
u"=D":"Laughing, big grin or laugh with glasses",
u"=3":"Laughing, big grin or laugh with glasses",
u"B\^D":"Laughing, big grin or laugh with glasses",
u":-\)\)":"Very happy",
u":-\(":"Frown, sad, andry or pouting",
u":-\(":"Frown, sad, andry or pouting",
u":\(":"Frown, sad, andry or pouting",
u":-c":"Frown, sad, andry or pouting",
u":c":"Frown, sad, andry or pouting",
u":-<":"Frown, sad, andry or pouting",
u":<":"Frown, sad, andry or pouting",
u":-\[":"Frown, sad, andry or pouting",
u":\[":"Frown, sad, andry or pouting",
u":-\|\|":"Frown, sad, andry or pouting",
u">:\[":"Frown, sad, andry or pouting",
u":\{":"Frown, sad, andry or pouting",
u":@":"Frown, sad, andry or pouting",
u">:\(":"Frown, sad, andry or pouting",
u":'-\(":"Crying",
u":'\(":"Crying",
u":'-\)":"Tears of happiness",
u":'\)":"Tears of happiness",
u"D-':":"Horror",
u"D:<":"Disgust",
u"D:":"Sadness",
u"D8":"Great dismay",
u"D;":"Great dismay",
u"D=":"Great dismay",
u"DX":"Great dismay",
u":-O":"Surprise",
u":O":"Surprise",
u":-o":"Surprise",
u":o":"Surprise",
u":-0":"Shock",
u"8-0":"Yawn",
u">:O":"Yawn",
u":-\*":"Kiss",
u":\*":"Kiss",
u":X":"Kiss",
u";-\)":"Wink or smirk",
u";\)":"Wink or smirk",
u"\*-\)":"Wink or smirk",
u"\*\)":"Wink or smirk",
u";-\]":"Wink or smirk",
u";\]":"Wink or smirk",
u";\^\)":"Wink or smirk",
u":-,":"Wink or smirk",
```

```
u";D":"Wink or smirk",
u":-P":"Tongue sticking out, cheeky, playful or blowing a raspberry",
u":P":"Tongue sticking out, cheeky, playful or blowing a raspberry",
u"X-P":"Tongue sticking out, cheeky, playful or blowing a raspberry",
u"XP":"Tongue sticking out, cheeky, playful or blowing a raspberry",
u":-Þ":"Tongue sticking out, cheeky, playful or blowing a raspberry",
u":Þ":"Tongue sticking out, cheeky, playful or blowing a raspberry",
u":b":"Tongue sticking out, cheeky, playful or blowing a raspberry",
u"d:":"Tongue sticking out, cheeky, playful or blowing a raspberry",
u"=p":"Tongue sticking out, cheeky, playful or blowing a raspberry",
u">:P":"Tongue sticking out, cheeky, playful or blowing a raspberry",
u":-/":"Skeptical, annoyed, undecided, uneasy or hesitant",
u":/":"Skeptical, annoyed, undecided, uneasy or hesitant",
u":-[.]":"Skeptical, annoyed, undecided, uneasy or hesitant",
u">:[(\\\)]":"Skeptical, annoyed, undecided, uneasy or hesitant",
u">:/":"Skeptical, annoyed, undecided, uneasy or hesitant",
u":[(\\\)]":"Skeptical, annoyed, undecided, uneasy or hesitant",
u"=/":"Skeptical, annoyed, undecided, uneasy or hesitant",
u"=[(\\\)]":"Skeptical, annoyed, undecided, uneasy or hesitant",
u":L":"Skeptical, annoyed, undecided, uneasy or hesitant",
u"=L":"Skeptical, annoyed, undecided, uneasy or hesitant",
u":S":"Skeptical, annoyed, undecided, uneasy or hesitant",
u":-\|":"Straight face",
u":\|":"Straight face",
u":$":"Embarrassed or blushing",
u":-x":"Sealed lips or wearing braces or tongue-tied",
u":x":"Sealed lips or wearing braces or tongue-tied",
u":-#":"Sealed lips or wearing braces or tongue-tied",
u":#":"Sealed lips or wearing braces or tongue-tied",
u":-&":"Sealed lips or wearing braces or tongue-tied",
u":&":"Sealed lips or wearing braces or tongue-tied",
u"O:-\)":"Angel, saint or innocent",
u"O:\)":"Angel, saint or innocent",
u"0:-3":"Angel, saint or innocent",
u"0:3":"Angel, saint or innocent",
u"0:-\)":"Angel, saint or innocent",
u"0:\)":"Angel, saint or innocent",
u":-b":"Tongue sticking out, cheeky, playful or blowing a raspberry",
u"0;\^\)":"Angel, saint or innocent",
u">:-\)":"Evil or devilish",
u">:\)":"Evil or devilish",
u"\}:-\)":"Evil or devilish",
u"\}:\)":"Evil or devilish",
u"3:-\)":"Evil or devilish",
u"3:\)":"Evil or devilish",
u">;\)":"Evil or devilish",
u"\|;-\)":"Cool",
u"\|-O":"Bored",
u":-J":"Tongue-in-cheek",
u"#-\)":"Party all night",
u"%-\)":"Drunk or confused",
u"%\)":"Drunk or confused",
u":-###..":"Being sick",
u":###..":"Being sick",
u"<:-\|":"Dump",
u"\(>_<\)":"Troubled",
u"\(>_<\)>":"Troubled",
u"\(^o^\)/":"Baby",
```

```
u"\( ; \) : Baby ",
u"\(\^\^>``":"Nervous or Embarrassed or Troubled or Shy or Sweat drop",
u"\(\^_\^;\)":"Nervous or Embarrassed or Troubled or Shy or Sweat drop",
u"\(-_-;\)":"Nervous or Embarrassed or Troubled or Shy or Sweat drop",
u"\(~_~;\) \( · \. · ;\)":"Nervous or Embarrassed or Troubled or Shy or Sweat drop",
u"\(-_-\)zzz":"Sleeping",
u"\(\^_-\)":"Wink",
u"\(\(\+_\+\)\)":"Confused",
u"\(\+o\+\)":"Confused",
u"\(o\|o\)":"Ultraman",
u"\^_\^":"Joyful",
u"\(\^_\^\)/":"Joyful",
u"\(\^O\^\)／":"Joyful",
u"\(\^o\^\)／":"Joyful",
u"\(__\)":"Kowtow as a sign of respect, or dogeza for apology",
u"_\(\._\.\)_":"Kowtow as a sign of respect, or dogeza for apology",
u"<\(_ _\)>":"Kowtow as a sign of respect, or dogeza for apology",
u"<m\(__\)m>":"Kowtow as a sign of respect, or dogeza for apology",
u"m\(__\)m":"Kowtow as a sign of respect, or dogeza for apology",
u"m\(_ _\)m":"Kowtow as a sign of respect, or dogeza for apology",
u"\('_'\)":"Sad or Crying",
u"\(/_;\)":"Sad or Crying",
u"\(T_T\) \(;_;\)":"Sad or Crying",
u"\(;_;":"Sad of Crying",
u"\(;_:\)":"Sad or Crying",
u"\(;O;\)":"Sad or Crying",
u"\(:_;\)":"Sad or Crying",
u"\(ToT\)":"Sad or Crying",
u";_;":"Sad or Crying",
u";-;":"Sad or Crying",
u";n;":"Sad or Crying",
u";;":"Sad or Crying",
u"Q\.Q":"Sad or Crying",
u"T\.T":"Sad or Crying",
u"QQ":"Sad or Crying",
u"Q_Q":"Sad or Crying",
u"\(-\.-\)":"Shame",
u"\(-_-\)":"Shame",
u"\(——\)":"Shame",
u"\(; —_—\)":"Shame",
u"\(=_=\)":"Tired",
u"\(=\^\·\^=\)":"cat",
u"\(=\^\·\·\^=\)":"cat",
u"=_\^= ":"cat",
u"\(\.\.\)":"Looking down",
u"\(\._\.\)":"Looking down",
u"\^m\^":"Giggling with hand covering mouth",
u"\(\ · \ · ?":"Confusion",
u"\(?_?\)":"Confusion",
u">\^_\^<":"Normal Laugh",
u"<\^!\^>":"Normal Laugh",
u"\^/\^":"Normal Laugh",
u"\ (\*\^_\^\*) " :"Normal Laugh",
u"\(\^<\^\) \(\^\.\^\)":"Normal Laugh",
u"\(^\^\)":"Normal Laugh",
u"\(\^\.\^\)":"Normal Laugh",
u"\(\^_\^\.\)":"Normal Laugh",
u"\(\^ \^\)":"Normal Laugh",
```

```
    u"\(\^\^\)":"Normal Laugh",
    u"\(\^J\^\)":"Normal Laugh",
    u"\(\*\^\.\^\*\)":"Normal Laugh",
    u"\(\^—\^\) ":"Normal Laugh",
    u"\(#\^\.\^#\)":"Normal Laugh",
    u"\ (\^—\^\) ":"Waving",
    u"\(;_;\)/~~~":"Waving",
    u"\(\^\.\^\)/~~~":"Waving",
    u"\(-_-\)/~~~ \($\·\·\)/~~~":"Waving",
    u"\(T_T\)/~~~":"Waving",
    u"\(ToT\)/~~~":"Waving",
    u"\(\*\^0\^\*\)":"Excited",
    u"\(\*_\*\)":"Amazed",
    u"\(\*_\*;":"Amazed",
    u"\(\+_\+\) \(@_@\)":"Amazed",
    u"\(\*\^\^\)v":"Laughing,Cheerful",
    u"\(\^_\^\)v":"Laughing,Cheerful",
    u"\(\(d[-_-]b\)\)":"Headphones,Listening to music",
    u'\(-"-\)':"Worried",
    u"\(——;\)":"Worried",
    u"\(\^0_0\^\)":"Eyeglasses",
    u"\(\^ v\^\)":"Happy",
    u"\(\^ u\^\)":"Happy",
    u"\(\^\)o\(\^\)":"Happy",
    u"\(\^O\^\)":"Happy",
    u"\(\^o\^\)":"Happy",
    u"\)\^o\^\(":"Happy",
    u":O o_O":"Surprised",
    u"o_0":"Surprised",
    u"o\.O":"Surpised",
    u"\(o\.o\)":"Surprised",
    u"oO":"Surprised",
    u"\(\*¯m¯\)":"Dissatisfied",
    u"\(‘A`\)":"Snubbed or Deflated"
}


EMOTICONS.update(EMOTICONS2)


!pip install emoji --upgrade
```

```
Collecting emoji
  Downloading https://files.pythonhosted.org/packages/40/8d/521be7f0091fe0f2ae690cc044faf4
     |████████████████████████████████| 51kB 2.0MB/s
Building wheels for collected packages: emoji
  Building wheel for emoji (setup.py) ... done
  Created wheel for emoji: filename=emoji-0.5.4-cp36-none-any.whl size=42175 sha256=c4fa1f
  Stored in directory: /root/.cache/pip/wheels/2a/a9/0a/4f8e8cce8074232aba240caca3fade315b
Successfully built emoji
Installing collected packages: emoji
Successfully installed emoji-0.5.4
```

```
import emoji


def convert_emoji(text):
  text = emoji.demojize(text)
```

```
  text = text.replace(":"," ")
  return text


##Function to replace the Emoticons
def convert_emoticons(text):
    for emot in EMOTICONS:
        text = text.replace(emot, EMOTICONS[emot])
    return text


## Functions to replace contractions
def replace_contractions(text):
    return contractions.fix(text)



def normalize_text (text):
  text = remove_URL(text)
  text = convert_emoticons(text)
  text = convert_emoji(text)
  text = replace_contractions(text)
  text = ' '.join(text.split())
  return text



Text_data.head()
```

	text	sentiment	confidence
0	trying to wait a patient as i can lol	1	0.3
1	Good morning Fotopro Team, I noticed in some ...	1	0.6
2	these still on track? ahead? behind?	1	0.2
3	any update on delivery?	1	0.0
4	product so nice had to get it twice	1	0.5

```
## Normalize the text using the normalize_text function

Text_data.iloc[:,0] = [normalize_text(j) for j in Text_data.iloc[:,0]]


Text_data.iloc[120,0]
```

```
'FYI if anyone wants a sholder strap pad I ended up getting the one located here'
```

```
# Remove the punctuations for FastText

table = str.maketrans(' ', ' ', string.punctuation)
Text_data.iloc[:,0] = [j.translate(table) for j in Text_data.iloc[:,0]]


# Remove double space
Text_data.iloc[:,0] = [" ".join(j.split()) for j in Text_data.iloc[:,0]]
```

```
frequent_stop = set(['the', 'a', 'an', 'and', 'to'])


def remove_stopwords(text):
    """custom function to remove the stopwords"""
    return " ".join([word for word in str(text).split() if word not in frequent_stop])


Text_data['text'] = Text_data['text'].apply(lambda text: remove_stopwords(text))

Text_data['text'] = Text_data['text'].str.lower()

Text_data.head()
```

	text	sentiment	confidence
0	trying wait patient as i can lol	1	0.3
1	good morning fotopro team i noticed in some of...	1	0.6
2	these still on track ahead behind	1	0.2
3	any update on delivery	1	0.0
4	product so nice had get it twice	1	0.5

Apply the Linear Transformation to Confidence Level

Since the confidence level ranged from 0 to 22, it is really hard to interpret its meaning. So we need to transform the range of confidence to [0,1]

```
old_min = Text_data['confidence'].min(axis=0)
old_max = Text_data['confidence'].max(axis=0)
new_min = 0
new_max = 1
Text_data['confidence'] = ( (Text_data['confidence'] - old_min) / (old_max - old_min) ) * (new_


Text_data['confidence'].describe()
```

```
count    204058.000000
mean          0.040472
std           0.040047
min           0.000000
25%           0.013453
50%           0.031390
75%           0.058296
max           1.000000
Name: confidence, dtype: float64
```

```
len(Text_data.loc[Text_data['confidence'] == 0].index)
```

```
22520
```

```
Undertermined = Text_data.loc[Text_data['confidence'] == 0]
```

```
## Remove those undertermined rows from traning & testing data

Useful_data = Text_data.drop(Text_data[Text_data['confidence'] == 0].index, axis = 0)


len(Useful_data) + len (Undertermined)
```

```
204058
```

Statistics of words

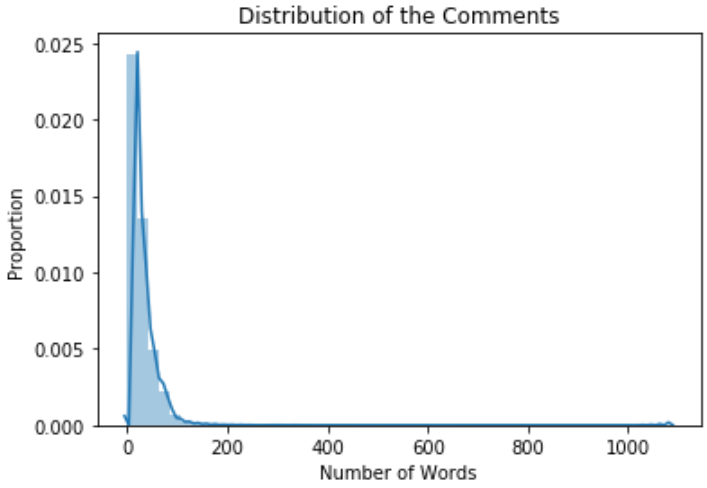
Most repreated words

```
from collections import Counter
cnt = Counter()
for text in Text_data['text'].values:
    for word in text.split():
        cnt[word] += 1

cnt.most_common(10)
```

```
[('i', 195186),
 ('you', 114632),
 ('is', 104459),
 ('it', 100900),
 ('for', 87645),
 ('of', 80481),
 ('not', 80328),
 ('have', 77570),
 ('in', 69203),
 ('my', 62892)]
```

```
import seaborn as sns
import numpy as np
%matplotlib inline
from scipy import stats

# Count maximum number of words per file.
wordDist = [len(w.split()) for w in Text_data.iloc[:,0]]
print('Avg. no of words: ' + str(np.round(np.mean(wordDist), 2)))
print('Std. deviation: ' + str(np.round(np.std(wordDist), 2)))
print('Max words: ' + str(np.max(wordDist)))
print('Min words: ' + str(np.min(wordDist)))

# Generate the plot
distCD = sns.distplot(wordDist)

# I'm saving the image to a PDF, as it makes it easier later to download.
distCD.figure.savefig("wordDist.pdf", format = "pdf")
plt.title('Distribution of the Comments')
plt.xlabel('Number of Words')
plt.ylabel('Proportion')
```

```
Avg. no of words: 27.56
Std. deviation: 25.47
Max words: 1084
Min words: 0
Text(14.375, 0.5, 'Proportion')
```



Embedding & Neural Network

```
tokenizer = Tokenizer() # Creates tokenizer model.
tokenizer.fit_on_texts(Text_data.iloc[:,0]) # Trains it over the tokens that we have.


# Import relevant packages
import os
import codecs

# Get words
Vals = list(tokenizer.word_index.keys())

# Write CSV with the output.
file = codecs.open('CrowdfundingWords.csv', "w", "utf-8")

for item in Vals:
    file.write("%s\r\n" % item)

file.close()


!ls
```

```
CrowdfundingWords.csv  'Full_Data (1).zip'   glove.42B.300d.zip   v0.9.1.zip
fastText-0.9.1          Full_Data.csv        sample_data          wordDist.pdf
```

```
!head CrowdfundingWords.csv
```

```
    i
    you
    is
    it
    for
    of
    not
    have
    in
    my
```

```
 Create word index from input
sed_sequences = tokenizer.texts_to_sequences(Useful_data.iloc[:,0]) # Create the sequences.
ndetermined_sequences = tokenizer.texts_to_sequences(Undertermined.iloc[:,0])

 Creates the indexes. Word index is a dictionary with words in it.
ord_index = tokenizer.word_index
rint('Found %s unique tokens.' % len(word_index))

 Creates the training dataset, adding padding when necessary.
sed_data = pad_sequences(used_sequences, maxlen=175,
                    padding = 'pre') # add padding at the end. No difference in practice.

ndetermined_data = pad_sequences(undetermined_sequences, maxlen=175,
                    padding = 'pre')

 Creates the objective function
sed_labels = Useful_data.iloc[:,1]
ndetermind_labels = Undertermined.iloc[:,1]

rint('Shape of used_data tensor:', used_data.shape)
rint('Shape of used_label tensor:', used_labels.shape)

rint('Shape of undetermined_data tensor:', undetermined_data.shape)
rint('Shape of undetermined_label tensor:', undetermind_labels.shape)
```

```
Found 108356 unique tokens.
Shape of used_data tensor: (181538, 175)
Shape of used_label tensor: (181538,)
Shape of undetermined_data tensor: (22520, 175)
Shape of undetermined_label tensor: (22520,)
```

```
# Create saving directory
!mkdir Preprocessed_data

# Save outputs
np.savetxt("Preprocessed_data/useful_Data.txt", used_data)
np.savetxt("Preprocessed_data/useful_Labels.txt", used_labels)
np.savetxt("Preprocessed_data/undetermined_Data.txt", undetermined_data)
np.savetxt("Preprocessed_data/undetermined_Label.txt", undetermind_labels)
```

Embedding with Glove

```
!ls
```

```
 CrowdfundingWords.csv   Full_Data.csv        sample_data
 fastText-0.9.1          glove.42B.300d.zip   v0.9.1.zip
'Full_Data (1).zip'      Preprocessed_data    wordDist.pdf
```

```
!unzip glove.42B.300d.zip
```

```
Archive:  glove.42B.300d.zip
  inflating: glove.42B.300d.txt
```

```
embeddings_index_Glove = {}
f = open(os.path.join('/content', 'glove.42B.300d.txt'))
for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index_Glove[word] = coefs
f.close()

print('Found %s word vectors.' % len(embeddings_index_Glove))
```

```
Found 1917494 word vectors.
```

```
len(word_index)
```

```
108356
```

```
embedding_matrix_Glove = np.zeros((len(word_index) + 1, 300))
for word, i in word_index.items():
    embedding_vector = embeddings_index_Glove.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix_Glove[i] = embedding_vector


embedding_matrix_Glove.shape
```

```
(108357, 300)
```

```
# Create saving directory
!mkdir GloveEmbed

# Save outputs
np.savetxt("GloveEmbed/embedding_matrix_Glove.txt", embedding_matrix_Glove)


!zip -r GloveEmbed.zip GloveEmbed
```

```
  adding: GloveEmbed/ (stored 0%)
  adding: GloveEmbed/embedding_matrix_Glove.txt (deflated 74%)
```

```
# Download files

from google.colab import files
files.download("GloveEmbed.zip")
```

Embedding with Fasttext

```
!wget https://dl.fbaipublicfiles.com/fasttext/vectors-english/crawl-300d-2M-subword.zip
```

```
--2019-12-16 17:21:25--  https://dl.fbaipublicfiles.com/fasttext/vectors-english/crawl-300
Resolving dl.fbaipublicfiles.com (dl.fbaipublicfiles.com)... 104.20.6.166, 104.20.22.166,
Connecting to dl.fbaipublicfiles.com (dl.fbaipublicfiles.com)|104.20.6.166|:443... connect
HTTP request sent, awaiting response... 200 OK
Length: 5828358084 (5.4G) [application/zip]
Saving to: 'crawl-300d-2M-subword.zip'

crawl-300d-2M-subwo 100%[===================>]   5.43G  13.2MB/s    in 6m 27s

2019-12-16 17:27:53 (14.4 MB/s) - 'crawl-300d-2M-subword.zip' saved [5828358084/5828358084
```

```
!unzip crawl-300d-2M-subword.zip
```

```
Archive:  crawl-300d-2M-subword.zip
  inflating: crawl-300d-2M-subword.vec
  inflating: crawl-300d-2M-subword.bin
```

```
!./fastText-0.9.1/fasttext print-sentence-vectors crawl-300d-2M-subword.bin < CrowdfundingWords
```

```
tcmalloc: large alloc 4800004096 bytes == 0x564c83ba0000 @  0x7f9bee7cc887 0x564c6fef68cf
tcmalloc: large alloc 2400002048 bytes == 0x564da1d44000 @  0x7f9bee7cc887 0x564c6fef68cf
```

```
!head EmbeddingFunding_Fast.tsv
```

```
0.018515 0.037727 -0.037647 -0.018836 -0.028652 0.043545 -0.030384 0.0056463 -0.055892 -0.
-0.0011453 -0.14067 0.10863 0.019816 0.039181 -0.08295 0.26816 0.032128 -0.053094 0.099946
0.0041258 0.16703 0.043264 -0.0037776 -0.0053039 -0.15283 0.13796 -0.0054034 -0.087125 -0.
0.011619 0.12207 0.1888 -0.0028967 -0.056286 -0.098217 0.16553 0.017025 -0.08861 0.12558 0
0.0073127 -0.069766 0.20653 0.035071 -0.036842 0.014323 0.29432 0.0077156 0.096258 -0.0367
0.012778 -0.028733 0.13043 0.038759 0.024786 -0.021207 0.19696 0.0090658 0.0054581 -0.0184
-0.0063514 0.084568 0.19908 -0.015154 0.045423 -0.17291 0.24113 -0.014154 -0.031689 -0.041
0.011582 -0.023062 0.15732 -0.0020891 0.008223 -0.1454 0.34851 -0.009148 -0.0067279 -0.037
0.017793 -0.16375 0.058917 0.011662 0.015755 0.016606 -0.075061 -0.00046485 -0.046867 -0.0
0.042654 -0.084826 0.041609 -0.045363 -0.019899 -0.026426 0.17565 0.009945 0.052309 0.1494
```

```
import fileinput

with fileinput.FileInput('EmbeddingFunding_Fast.tsv', inplace=True, backup='.bak') as file:
    for line in file:
        print(line.replace(' ', ','), end='')
```

```
import numpy as np
import os

# Create the first line
firstLine = ','.join(['D'+str(i) for i in np.arange(1, 301)]) + '\n'

# Open as read only. Read the file
```

```
# Open as read only. Read the file
with open('EmbeddingFunding_Fast.tsv', 'r') as original:
  data = original.read()

# Open to write and write the first line and the rest
with open('EmbeddingFunding_Fast.csv', 'w') as modified:
  modified.write(firstLine + data)


!head EmbeddingFunding_Fast.csv
```

```
D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,D16,D17,D18,D19,D20,D21,D22,D23,D24,D25
0.018515,0.037727,-0.037647,-0.018836,-0.028652,0.043545,-0.030384,0.0056463,-0.055892,-0.
-0.0011453,-0.14067,0.10863,0.019816,0.039181,-0.08295,0.26816,0.032128,-0.053094,0.099946
0.0041258,0.16703,0.043264,-0.0037776,-0.0053039,-0.15283,0.13796,-0.0054034,-0.087125,-0.
0.011619,0.12207,0.1888,-0.0028967,-0.056286,-0.098217,0.16553,0.017025,-0.08861,0.12558,0
0.0073127,-0.069766,0.20653,0.035071,-0.036842,0.014323,0.29432,0.0077156,0.096258,-0.0367
0.012778,-0.028733,0.13043,0.038759,0.024786,-0.021207,0.19696,0.0090658,0.0054581,-0.0184
-0.0063514,0.084568,0.19908,-0.015154,0.045423,-0.17291,0.24113,-0.014154,-0.031689,-0.041
0.011582,-0.023062,0.15732,-0.0020891,0.008223,-0.1454,0.34851,-0.009148,-0.0067279,-0.037
0.017793,-0.16375,0.058917,0.011662,0.015755,0.016606,-0.075061,-0.00046485,-0.046867,-0.0
```

```
# Read word embeddings
Embeddings_Fasttext= pd.read_csv('EmbeddingFunding_Fast.csv', sep=',', decimal = '.',
                         low_memory = True, index_col = False)
Embeddings_Fasttext.describe()
```

	D1	D2	D3	D4	D5	D6	
count	108356.000000	108356.000000	108356.000000	108356.000000	108356.000000	108356.000000	108
mean	-0.019270	-0.019756	0.060673	0.009534	-0.009059	-0.070565	
std	0.036123	0.073797	0.064230	0.046206	0.045283	0.073810	
min	-0.231400	-0.454080	-0.280250	-0.222680	-0.231170	-0.401990	
25%	-0.040008	-0.060709	0.017764	-0.019577	-0.038647	-0.115530	
50%	-0.016764	-0.012729	0.061215	0.011206	-0.009610	-0.068437	
75%	0.003494	0.027083	0.103373	0.040535	0.020074	-0.028587	
max	0.248060	0.339740	0.359800	0.227040	0.211800	0.331110	

8 rows × 300 columns

```
# Create embedding dictionary

EmbeddingsDict_Fast = dict(zip(Vals, Embeddings_Fasttext.values))

used_data[0]
```

```
array([  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 342,
       123, 973,  21,   1,  20, 937], dtype=int32)
```

```
# Create first matrix full with 0's
embedding_matrix_Fast = np.zeros((len(word_index) + 1, 300))

# Generate embeddings matrix
for word, i in word_index.items():
    embedding_vector = EmbeddingsDict_Fast.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix_Fast[i] = embedding_vector

# Print what came out
embedding_matrix_Fast
```

```
array([[ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.018515  ,  0.037727  , -0.037647  , ..., -0.0037811 ,
         0.02121   , -0.0054264 ],
       [-0.0011453 , -0.14067   ,  0.10863   , ..., -0.04363   ,
         0.010488  , -0.016025  ],
       ...,
       [-0.060361  ,  0.07252   ,  0.0026688 , ..., -0.052618  ,
        -0.0019166 , -0.0031732 ],
       [ 0.013113  , -0.12469   ,  0.036819  , ...,  0.0073507 ,
        -0.00078115, -0.030591  ],
       [ 0.0020908 , -0.24629   ,  0.099815  , ...,  0.0083639 ,
         0.013606  , -0.030303  ]])
```

```
embedding_matrix_Fast.shape
```

```
(108357, 300)
```

```
# Create saving directory
!mkdir FastEmbed

# Save outputs
np.savetxt("FastEmbed/embedding_matrix_Fast", embedding_matrix_Fast)
```

```
!zip -r FastEmbed.zip FastEmbed
```

```
  adding: FastEmbed/ (stored 0%)
  adding: FastEmbed/embedding_matrix_Fast (deflated 75%)
```

```
# Download files

from google.colab import files
files.download("FastEmbed.zip")


# We will also save the word dictionary
# A pickle file is a Python native file
import pickle
f = open("WordDictionary.pkl","wb") #write in binary mode
pickle.dump(word_index, f)
f.close()


# Zip all files for download.

!zip -r Preprocessed_data.zip Preprocessed_data


# Download files

from google.colab import files
files.download("Preprocessed_data.zip")
```

Build the Architectures of the Neural Network

we first split the data into training and test set

```
from sklearn.model_selection import train_test_split


# Split into train and test
X_train, X_test, y_train, y_test = train_test_split(used_data, used_labels,
                                                    test_size=0.33,
                                                    random_state=251121253,
                                                    stratify = used_labels)
```

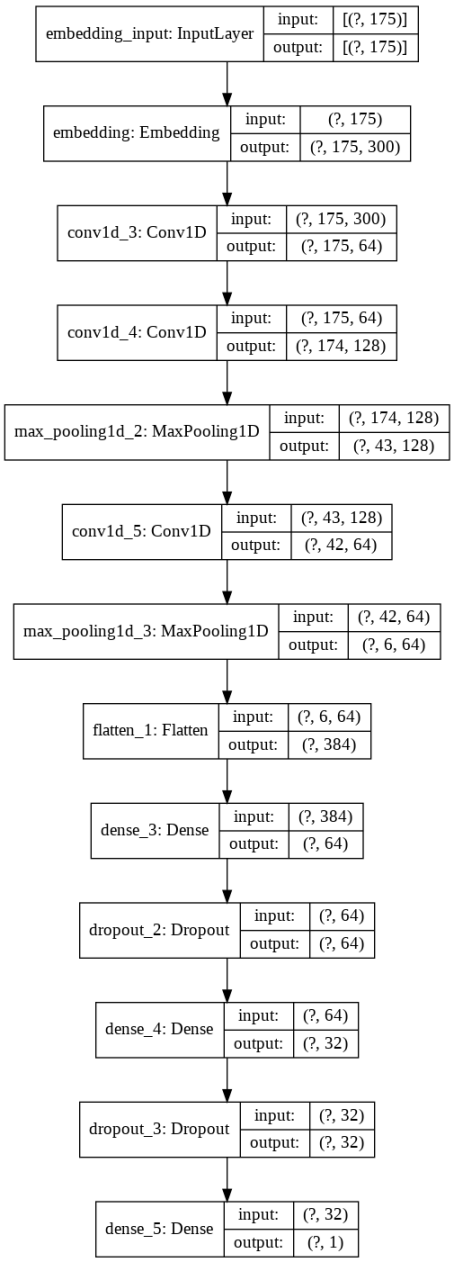
My own sequential structure

```
Sequential_model = Sequential()

embedding_layer_Glove = Embedding(len(word_index) + 1,
                            300,
                            weights=[embedding_matrix_Glove],
                            input_length=175,
                            trainable=False)

embedding_layer_Fast = Embedding(len(word_index) + 1,
                            300,
                            weights=[embedding_matrix_Fast],
                            input_length=175,
                            trainable=False)


Sequential_model.add(embedding_layer_Glove)
```

```
Sequential_model.add(Conv1D(filters=64,
                 kernel_size=1,
                 padding='valid',
                 activation='relu',
                 strides=1)
         )

Sequential_model.add(Conv1D(filters=128,
                 kernel_size=2,
                 padding='valid',
                 activation='relu',
                 strides=1)
         )

Sequential_model.add(MaxPooling1D(pool_size = 4))

Sequential_model.add(Conv1D(filters=64,
                 kernel_size=2,
                 padding='valid',
                 activation='relu',
                 strides=1)
         )

Sequential_model.add(MaxPooling1D(pool_size = 7))

# Flatten
Sequential_model.add(Flatten())



Sequential_model.add(Dense(64, activation = 'relu'))
Sequential_model.add(Dropout(0.5))

Sequential_model.add(Dense(32, activation = 'relu'))
Sequential_model.add(Dropout(0.5))


# Output layer of size 1
Sequential_model.add(Dense(1, activation = 'sigmoid'))


Sequential_model.summary()
```

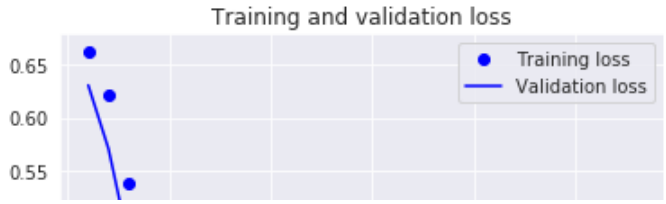
```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 175, 300)          32507100
_________________________________________________________________
conv1d_9 (Conv1D)            (None, 175, 64)           19264
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 174, 128)          16512
_________________________________________________________________
max_pooling1d_6 (MaxPooling1 (None, 43, 128)           0
_________________________________________________________________
conv1d_11 (Conv1D)           (None, 42, 64)            16448
_________________________________________________________________
max_pooling1d_7 (MaxPooling1 (None, 6, 64)             0
_________________________________________________________________
flatten_3 (Flatten)          (None, 384)               0
_________________________________________________________________
dense_9 (Dense)              (None, 64)                24640
_________________________________________________________________
dropout_6 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_10 (Dense)             (None, 32)                2080
_________________________________________________________________
dropout_7 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 33
=================================================================
Total params: 32,586,077
Trainable params: 78,977
Non-trainable params: 32,507,100
_________________________________________________________________
```

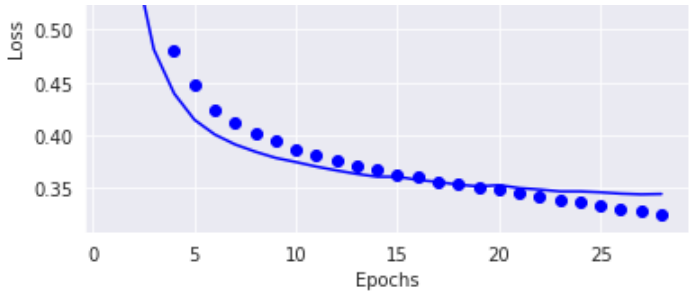
```
# Use Adam as optimizer, with a binary_crossentropy error.
adam = keras.optimizers.Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999, amsgrad=False)


Sequential_model.compile(loss='binary_crossentropy',
              optimizer=adam,
              metrics=['acc'])


import matplotlib.pyplot as plt
from tensorflow.keras.utils import plot_model
from IPython.display import Image
%matplotlib inline

plot_model(Sequential_model, show_shapes=True, show_layer_names=True, to_file='model.png')
Image(retina=True, filename='model.png')
```




```
it the model
tory = Sequential_model.fit(X_train, y_train, validation_split=0.33, epochs=28, batch_size=1000
.set_style("darkgrid")
s = history.history['loss']
_loss = history.history['val_loss']
chs = range(1, len(loss) + 1)
.plot(epochs, loss, 'bo', label='Training loss')
.plot(epochs, val_loss, 'b', label='Validation loss')
.title('Training and validation loss')
.xlabel('Epochs')
.ylabel('Loss')
.legend()
.show()
```

```
Train on 81492 samples, validate on 40138 samples
Epoch 1/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.6618 - acc: 0.629
Epoch 2/28
81492/81492 [==============================] - 8s 97us/sample - loss: 0.6215 - acc: 0.6709
Epoch 3/28
81492/81492 [==============================] - 8s 99us/sample - loss: 0.5373 - acc: 0.7030
Epoch 4/28
81492/81492 [==============================] - 8s 98us/sample - loss: 0.4793 - acc: 0.7596
Epoch 5/28
81492/81492 [==============================] - 8s 98us/sample - loss: 0.4474 - acc: 0.7855
Epoch 6/28
81492/81492 [==============================] - 8s 99us/sample - loss: 0.4247 - acc: 0.8015
Epoch 7/28
81492/81492 [==============================] - 8s 99us/sample - loss: 0.4115 - acc: 0.8096
Epoch 8/28
81492/81492 [==============================] - 8s 99us/sample - loss: 0.4016 - acc: 0.8164
Epoch 9/28
81492/81492 [==============================] - 8s 100us/sample - loss: 0.3950 - acc: 0.820
Epoch 10/28
81492/81492 [==============================] - 8s 100us/sample - loss: 0.3863 - acc: 0.825
Epoch 11/28
81492/81492 [==============================] - 8s 100us/sample - loss: 0.3811 - acc: 0.828
Epoch 12/28
81492/81492 [==============================] - 8s 100us/sample - loss: 0.3770 - acc: 0.830
Epoch 13/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3714 - acc: 0.833
Epoch 14/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3674 - acc: 0.835
Epoch 15/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3633 - acc: 0.838
Epoch 16/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3615 - acc: 0.840
Epoch 17/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3566 - acc: 0.841
Epoch 18/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3536 - acc: 0.844
Epoch 19/28
81492/81492 [==============================] - 8s 100us/sample - loss: 0.3503 - acc: 0.845
Epoch 20/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3491 - acc: 0.845
Epoch 21/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3452 - acc: 0.847
Epoch 22/28
81492/81492 [==============================] - 8s 100us/sample - loss: 0.3420 - acc: 0.848
Epoch 23/28
81492/81492 [==============================] - 8s 100us/sample - loss: 0.3393 - acc: 0.850
Epoch 24/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3369 - acc: 0.852
Epoch 25/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3333 - acc: 0.853
Epoch 26/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3302 - acc: 0.855
Epoch 27/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3283 - acc: 0.857
Epoch 28/28
81492/81492 [==============================] - 8s 100us/sample - loss: 0.3254 - acc: 0.858
```

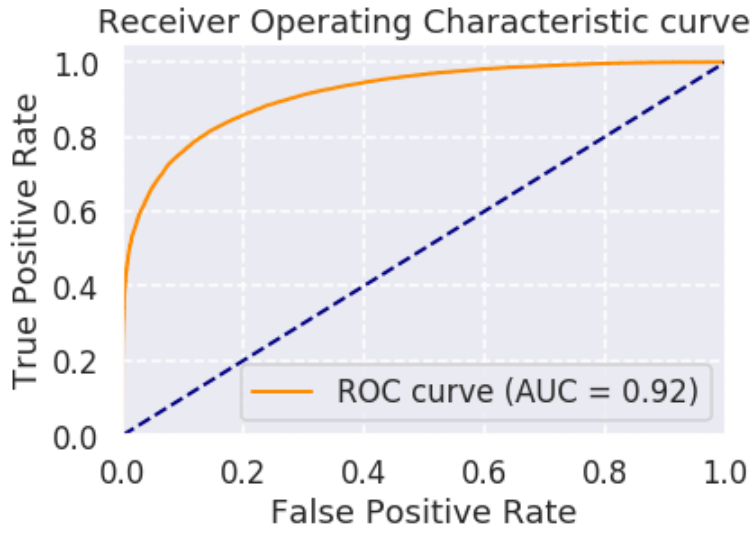





```
# Calculate outputs in test set
prob_test_GS = Sequential_model.predict(X_test, verbose = 1)
prob_train = Sequential_model.predict(X_train, verbose = 1)

# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(y_train, prob_train)
roc_auc = auc(fpr, tpr)
print('\nAUC train: ', roc_auc)

# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(y_test, prob_test_GS)
roc_auc = auc(fpr, tpr)
print('AUC test: ', roc_auc)
```

```
59908/59908 [==============================] - 6s 98us/sample
121630/121630 [==============================] - 12s 99us/sample

AUC train:  0.9277027624282317
AUC test:  0.9155071780293197
```

```
sns.set('talk', 'darkgrid', 'dark', font_scale=1, \
        rc={"lines.linewidth": 2, 'grid.linestyle': '--'})

lw = 2
plt.figure()
plt.plot(fpr, tpr, color='darkorange',
         lw=lw, label='ROC curve (AUC = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic curve')
plt.legend(loc="lower right")

# I am saving the output as a PDF for easy exporting.
plt.savefig('roc_auc.pdf', format = "pdf")

# Now I show the plot inline.
plt.show()
```




```
Cutoff = 0.5
```

```
prob_test_GS[prob_test_GS > Cutoff] = 1
prob_test_GS[prob_test_GS <= Cutoff] = 0
```

```
from sklearn.metrics import confusion_matrix
```

```
confusion_matrix1 = \
confusion_matrix(y_true = y_test, y_pred = prob_test_GS)
```

```
confusion_matrix1
```

```
array([[14758,  4840],
       [ 4596, 35714]])
```

```
len(y_test[y_test == 0] )
```

```
19598
```

```
accuracy = (confusion_matrix1[0][0] + confusion_matrix1[1][1])/ \
(confusion_matrix1[0][0] + confusion_matrix1[0][1] + \
 confusion_matrix1[1][0] + confusion_matrix1[1][1])
```

```
accuracy
```

```
0.8424918207918809
```

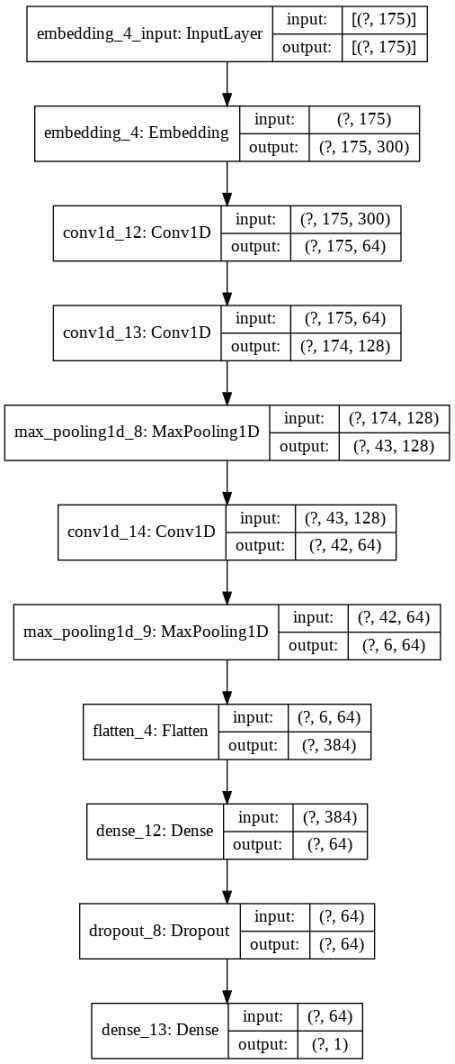
```
Sequential_model_FT = Sequential()
```

```
embedding_layer_Fast = Embedding(len(word_index) + 1,
                            300,
                            weights=[embedding_matrix_Fast],
                            input_length=175,
                            trainable=False)
```

```
Sequential_model_FT.add(embedding_layer_Fast)

Sequential_model_FT.add(Conv1D(filters=64,
                        kernel_size=1,
                        padding='valid',
                        activation='relu',
                        strides=1)
         )

Sequential_model_FT.add(Conv1D(filters=128,
                 kernel_size=2,
                 padding='valid',
                 activation='relu',
                 strides=1)
         )

Sequential_model_FT.add(MaxPooling1D(pool_size = 4))

Sequential_model_FT.add(Conv1D(filters=64,
                 kernel_size=2,
                 padding='valid',
                 activation='relu',
                 strides=1)
         )

Sequential_model_FT.add(MaxPooling1D(pool_size = 7))

# Flatten
Sequential_model_FT.add(Flatten())

# Dense Layer of size 128 with Dropout
Sequential_model_FT.add(Dense(64, activation = 'relu'))
Sequential_model_FT.add(Dropout(0.5))

# Output layer of size 1
Sequential_model_FT.add(Dense(1, activation = 'sigmoid'))

Sequential_model_FT.summary()
```

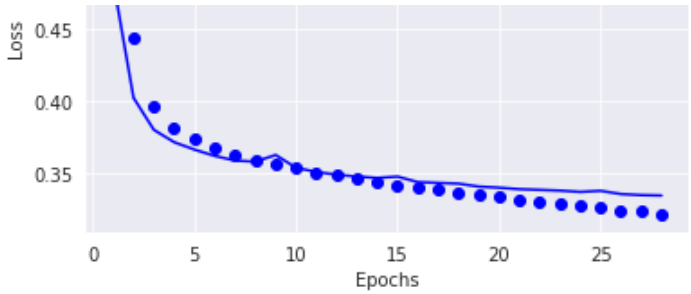
```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_5 (Embedding)      (None, 175, 300)          32507100  
_________________________________________________________________
conv1d_15 (Conv1D)           (None, 175, 64)           19264     
_________________________________________________________________
conv1d_16 (Conv1D)           (None, 174, 128)          16512     
_________________________________________________________________
max_pooling1d_10 (MaxPooling (None, 43, 128)           0         
_________________________________________________________________
conv1d_17 (Conv1D)           (None, 42, 64)            16448     
_________________________________________________________________
max_pooling1d_11 (MaxPooling (None, 6, 64)             0         
_________________________________________________________________
flatten_5 (Flatten)          (None, 384)               0         
_________________________________________________________________
dense_14 (Dense)             (None, 64)                24640     
_________________________________________________________________
dropout_9 (Dropout)          (None, 64)                0         
_________________________________________________________________
dense_15 (Dense)             (None, 1)                 65        
=================================================================
Total params: 32,584,029
Trainable params: 76,929
Non-trainable params: 32,507,100
_________________________________________________________________
```

```
# Use Adam as optimizer, with a binary_crossentropy error.
adam = keras.optimizers.Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999, amsgrad=False)


Sequential_model_FT.compile(loss='binary_crossentropy',
              optimizer=adam,
              metrics=['acc'])


port matplotlib.pyplot as plt
om tensorflow.keras.utils import plot_model
om IPython.display import Image
atplotlib inline

ot_model(Sequential_model_FT, show_shapes=True, show_layer_names=True, to_file='model.png')
age(retina=True, filename='model.png')
```




```
# Fit the model
history = Sequential_model_FT.fit(X_train, y_train, validation_split=0.33, epochs=28, batch_siz
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(loss) + 1)
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

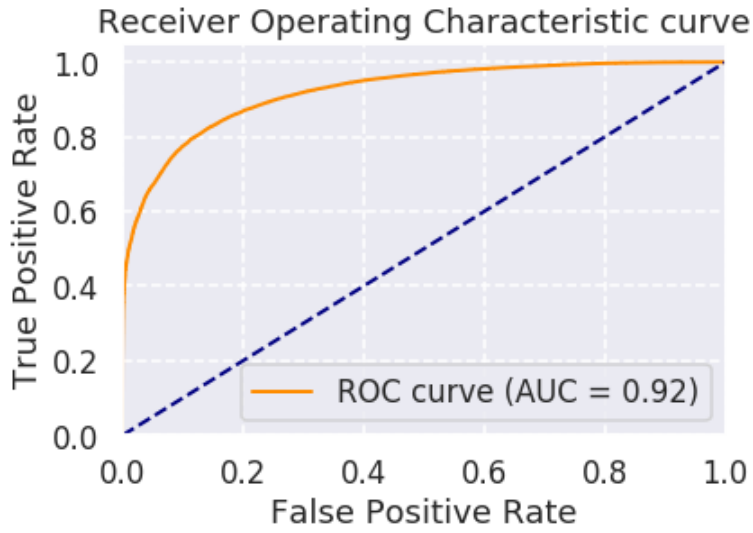
```
Train on 81492 samples, validate on 40138 samples
Epoch 1/28
81492/81492 [==============================] - 9s 109us/sample - loss: 0.5698 - acc: 0.688
Epoch 2/28
81492/81492 [==============================] - 8s 104us/sample - loss: 0.4449 - acc: 0.785
Epoch 3/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3971 - acc: 0.814
Epoch 4/28
81492/81492 [==============================] - 8s 103us/sample - loss: 0.3810 - acc: 0.825
Epoch 5/28
81492/81492 [==============================] - 8s 102us/sample - loss: 0.3738 - acc: 0.828
Epoch 6/28
81492/81492 [==============================] - 8s 104us/sample - loss: 0.3673 - acc: 0.833
Epoch 7/28
81492/81492 [==============================] - 9s 105us/sample - loss: 0.3626 - acc: 0.835
Epoch 8/28
81492/81492 [==============================] - 8s 103us/sample - loss: 0.3589 - acc: 0.837
Epoch 9/28
81492/81492 [==============================] - 8s 103us/sample - loss: 0.3564 - acc: 0.838
Epoch 10/28
81492/81492 [==============================] - 8s 103us/sample - loss: 0.3533 - acc: 0.839
Epoch 11/28
81492/81492 [==============================] - 8s 102us/sample - loss: 0.3505 - acc: 0.841
Epoch 12/28
81492/81492 [==============================] - 9s 106us/sample - loss: 0.3483 - acc: 0.843
Epoch 13/28
81492/81492 [==============================] - 8s 102us/sample - loss: 0.3460 - acc: 0.843
Epoch 14/28
81492/81492 [==============================] - 8s 103us/sample - loss: 0.3441 - acc: 0.844
Epoch 15/28
81492/81492 [==============================] - 8s 102us/sample - loss: 0.3416 - acc: 0.846
Epoch 16/28
81492/81492 [==============================] - 8s 103us/sample - loss: 0.3396 - acc: 0.847
Epoch 17/28
81492/81492 [==============================] - 8s 102us/sample - loss: 0.3380 - acc: 0.847
Epoch 18/28
81492/81492 [==============================] - 8s 102us/sample - loss: 0.3363 - acc: 0.848
Epoch 19/28
81492/81492 [==============================] - 8s 102us/sample - loss: 0.3347 - acc: 0.849
Epoch 20/28
81492/81492 [==============================] - 8s 101us/sample - loss: 0.3331 - acc: 0.850
Epoch 21/28
81492/81492 [==============================] - 9s 107us/sample - loss: 0.3308 - acc: 0.852
Epoch 22/28
81492/81492 [==============================] - 8s 103us/sample - loss: 0.3302 - acc: 0.852
Epoch 23/28
81492/81492 [==============================] - 8s 102us/sample - loss: 0.3281 - acc: 0.853
Epoch 24/28
81492/81492 [==============================] - 8s 103us/sample - loss: 0.3272 - acc: 0.854
Epoch 25/28
81492/81492 [==============================] - 8s 102us/sample - loss: 0.3255 - acc: 0.854
Epoch 26/28
81492/81492 [==============================] - 9s 106us/sample - loss: 0.3241 - acc: 0.856
Epoch 27/28
81492/81492 [==============================] - 8s 103us/sample - loss: 0.3233 - acc: 0.856
Epoch 28/28
81492/81492 [==============================] - 8s 104us/sample - loss: 0.3211 - acc: 0.856
```




```
# Calculate outputs in test set
prob_test_FS = Sequential_model_FT.predict(X_test, verbose = 1)
prob_train = Sequential_model_FT.predict(X_train, verbose = 1)

# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(y_train, prob_train)
roc_auc = auc(fpr, tpr)
print('\nAUC train: ', roc_auc)

# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(y_test, prob_test_FS)
roc_auc = auc(fpr, tpr)
print('AUC test: ', roc_auc)
```

```
59908/59908 [==============================] - 6s 94us/sample
121630/121630 [==============================] - 11s 94us/sample

AUC train:  0.9271910063294806
AUC test:  0.92077067020316
```

```
sns.set('talk', 'darkgrid', 'dark', font_scale=1, \
        rc={"lines.linewidth": 2, 'grid.linestyle': '--'})

lw = 2
plt.figure()
plt.plot(fpr, tpr, color='darkorange',
         lw=lw, label='ROC curve (AUC = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic curve')
plt.legend(loc="lower right")

# I am saving the output as a PDF for easy exporting.
plt.savefig('roc_auc.pdf', format = "pdf")

# Now I show the plot inline.
plt.show()
```




```
prob_test_FS[prob_test_FS > Cutoff] = 1
prob_test_FS[prob_test_FS <= Cutoff] = 0
```

```
confusion_matrix2 = \
confusion_matrix(y_true = y_test, y_pred = prob_test_FS)
```

```
confusion_matrix2
```

```
array([[14679,  4919],
       [ 4159, 36151]])
```

```
# Calculate outputs in test set
prob_test2 = Sequential_model2.predict(undetermined_data, verbose = 1)
```

```
# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(undetermind_labels, prob_test2)
roc_auc = auc(fpr, tpr)
```

```
22520/22520 [==============================] - 1s 54us/sample
/usr/local/lib/python3.6/dist-packages/sklearn/metrics/ranking.py:651: UndefinedMetricWarn
  UndefinedMetricWarning)
/usr/local/lib/python3.6/dist-packages/sklearn/metrics/ranking.py:113: RuntimeWarning: inv
  if np.any(dx < 0):
```

```
prob_test2
```

```
array([[0.62579656],
       [0.945838  ],
       [0.46042684],
       ...,
       [0.9191065 ],
       [0.9688797 ],
       [0.5518097 ]], dtype=float32)
```

```
len(prob_test2[prob_test2 <0.3]) + len(prob_test2[prob_test2 > 0.7])
```

```
15338
```

```
len(prob_test2)
```

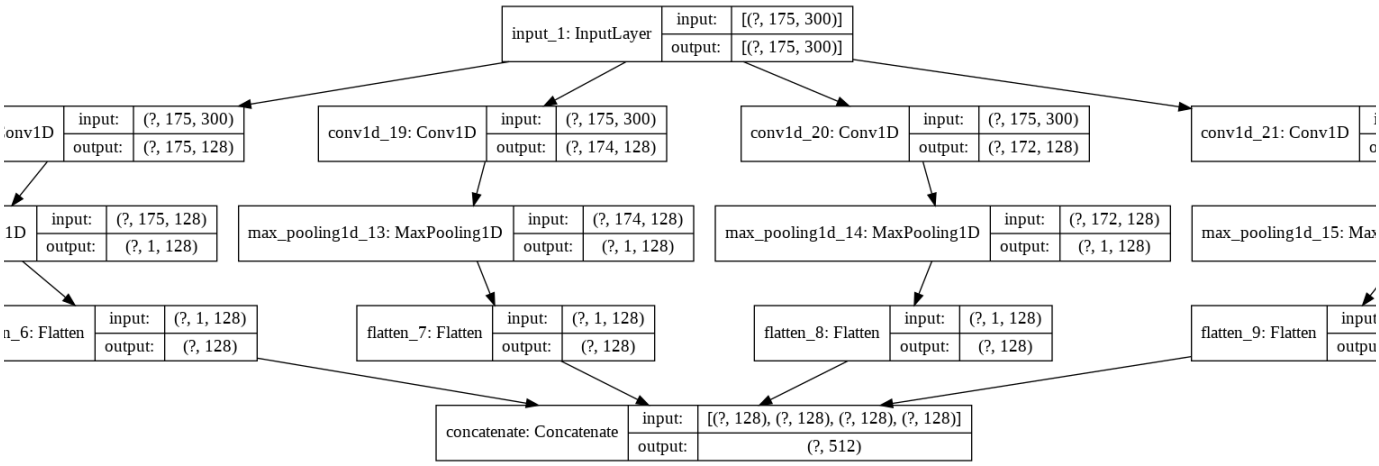
```
22520
```

Kim's Structure

```
sgd = keras.optimizers.SGD(learning_rate=0.01, momentum=0.0, nesterov=False)


# Filter sizes to use.
filter_sizes = (1,2,4,5)

# Initialize. We need to give it the input dimension (from the Embedding!)
graph_in = Input(shape=(175, 300))
convs = []
avgs = []

# This for stacks the layers. Inside each for, we build the sequence of layer. The command "app
# that to the "conv" variable, which is simply a stack of convolutions.
for fsz in filter_sizes:
    conv = Conv1D(filters=128,
                  kernel_size=fsz,
                  padding='valid',
                  activation='relu',
                  strides=1)(graph_in)

    pool = MaxPooling1D(pool_size= 175 - fsz + 1)(conv) # Put this layer AFTER the convolution
    flattenMax = Flatten()(pool) # Flatten the pooling layer.
    convs.append(flattenMax) # Append this to the convs object that saves the stack.

# Concatenate layers.
if len(filter_sizes)>1:
    out = Concatenate()(convs)
else:
    out = convs[0]

graph = Model(inputs=graph_in, outputs=out, name="graphModel")

graph.summary()
```

```
Model: "graphModel"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 175, 300)]   0
__________________________________________________________________________________________________
conv1d_18 (Conv1D)              (None, 175, 128)     38528       input_1[0][0]
__________________________________________________________________________________________________
conv1d_19 (Conv1D)              (None, 174, 128)     76928       input_1[0][0]
__________________________________________________________________________________________________
conv1d_20 (Conv1D)              (None, 172, 128)     153728      input_1[0][0]
__________________________________________________________________________________________________
conv1d_21 (Conv1D)              (None, 171, 128)     192128      input_1[0][0]
__________________________________________________________________________________________________
max_pooling1d_12 (MaxPooling1D) (None, 1, 128)       0           conv1d_18[0][0]
__________________________________________________________________________________________________
max_pooling1d_13 (MaxPooling1D) (None, 1, 128)       0           conv1d_19[0][0]
__________________________________________________________________________________________________
max_pooling1d_14 (MaxPooling1D) (None, 1, 128)       0           conv1d_20[0][0]
__________________________________________________________________________________________________
max_pooling1d_15 (MaxPooling1D) (None, 1, 128)       0           conv1d_21[0][0]
__________________________________________________________________________________________________
flatten_6 (Flatten)             (None, 128)          0           max_pooling1d_12[0][0]
__________________________________________________________________________________________________
flatten_7 (Flatten)             (None, 128)          0           max_pooling1d_13[0][0]
__________________________________________________________________________________________________
flatten_8 (Flatten)             (None, 128)          0           max_pooling1d_14[0][0]
__________________________________________________________________________________________________
flatten_9 (Flatten)             (None, 128)          0           max_pooling1d_15[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 512)          0           flatten_6[0][0]
                                                                 flatten_7[0][0]
                                                                 flatten_8[0][0]
                                                                 flatten_9[0][0]
==================================================================================================
Total params: 461,312
Trainable params: 461,312
Non-trainable params: 0
__________________________________________________________________________________________________
```

```
import matplotlib.pyplot as plt
from tensorflow.keras.utils import plot_model
from IPython.display import Image
%matplotlib inline

plot_model(graph, show_shapes=True, show_layer_names=True, to_file='GraphModel.png')
Image(retina=True, filename='GraphModel.png')
```

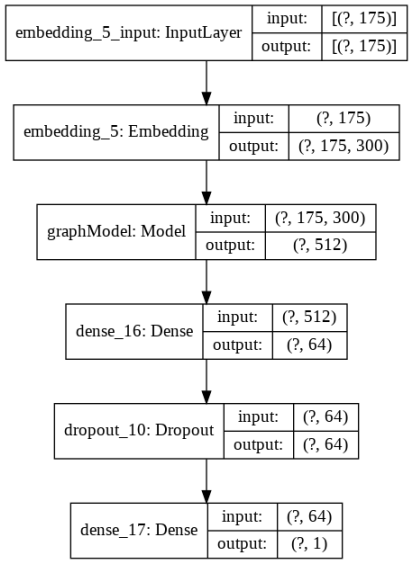



```
# Final model
Kim = Sequential()

Kim.add(embedding_layer_Fast)

# Now we add our graph model
Kim.add(graph)

# Add a few layers
Kim.add(Dense(64, activation='relu'))
Kim.add(Dropout(0.5))
Kim.add(Dense(1, activation='sigmoid'))

# adam = Adam(clipnorm=.1)
Kim.compile(loss='binary_crossentropy',
              optimizer=adam,
              metrics=['acc'])


Kim.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_5 (Embedding)      (None, 175, 300)          32507100  
_________________________________________________________________
graphModel (Model)           (None, 512)               461312    
_________________________________________________________________
dense_16 (Dense)             (None, 64)                32832     
_________________________________________________________________
dropout_10 (Dropout)         (None, 64)                0         
_________________________________________________________________
dense_17 (Dense)             (None, 1)                 65        
=================================================================
Total params: 33,001,309
Trainable params: 494,209
Non-trainable params: 32,507,100
_________________________________________________________________
```

```
import matplotlib.pyplot as plt
from tensorflow.keras.utils import plot_model
```

```
from IPython.display import Image
%matplotlib inline

plot_model(Kim, show_shapes=True, show_layer_names=True, to_file='model.png')
Image(retina=True, filename='model.png')
```



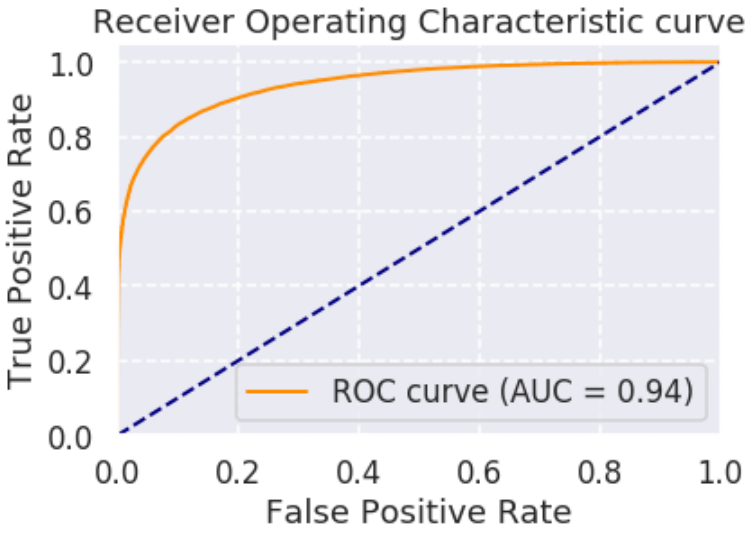

```
history = Kim.fit(X_train, y_train, validation_split=0.33, epochs=9, batch_size=600)
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(loss) + 1)
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
2 samples, validate on 40138 samples

==============================] - 57s 694us/sample - loss: 0.4888 - acc: 0.7474 - val_loss:

==============================] - 55s 678us/sample - loss: 0.3735 - acc: 0.8370 - val_loss:

==============================] - 56s 685us/sample - loss: 0.3382 - acc: 0.8542 - val_loss:

==============================] - 56s 682us/sample - loss: 0.3210 - acc: 0.8629 - val_loss:

==============================] - 56s 686us/sample - loss: 0.3050 - acc: 0.8707 - val_loss:

==============================] - 56s 682us/sample - loss: 0.2954 - acc: 0.8758 - val_loss:

==============================] - 56s 683us/sample - loss: 0.2846 - acc: 0.8820 - val_loss:

==============================] - 55s 673us/sample - loss: 0.2762 - acc: 0.8859 - val_loss:

==============================] - 55s 681us/sample - loss: 0.2678 - acc: 0.8906 - val_loss:
```


```
# Calculate outputs in test set
prob_test_FK = Kim.predict(X_test, verbose = 1)
prob_train = Kim.predict(X_train, verbose = 1)

# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(y_train, prob_train)
roc_auc = auc(fpr, tpr)
print('\nAUC train: ', roc_auc)

# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(y_test, prob_test_FK)
roc_auc = auc(fpr, tpr)
print('AUC test: ', roc_auc)

sns.set('talk', 'darkgrid', 'dark', font_scale=1, \
        rc={"lines.linewidth": 2, 'grid.linestyle': '--'})

lw = 2
plt.figure()
plt.plot(fpr, tpr, color='darkorange',
         lw=lw, label='ROC curve (AUC = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
```

```
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic curve')
plt.legend(loc="lower right")
plt.savefig('roc_auc.pdf', format = "pdf")
plt.show()
```

```
59908/59908 [==============================] - 14s 240us/sample
121630/121630 [==============================] - 29s 239us/sample

AUC train:  0.9532259079751589
AUC test:  0.9417653056401418
```



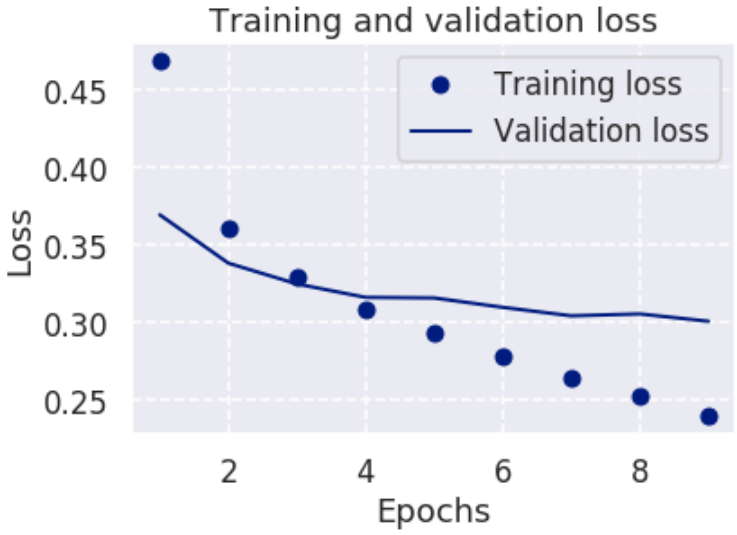

```
prob_test_FK[prob_test_FK > Cutoff] = 1
prob_test_FK[prob_test_FK <= Cutoff] = 0


confusion_matrix3 = \
confusion_matrix(y_true = y_test, y_pred = prob_test_FK)


confusion_matrix3
```

```
array([[15506,  4092],
       [ 3658, 36652]])
```

```
Kim_G = Sequential()

Kim_G.add(embedding_layer_Glove)

# Now we add our graph model
Kim_G.add(graph)

# Add a few layers
Kim_G.add(Dense(64, activation='relu'))
Kim_G.add(Dropout(0.5))
Kim_G.add(Dense(1, activation='sigmoid'))

# adam = Adam(clipnorm=.1)
Kim_G.compile(loss='binary_crossentropy',
```

```
                optimizer=adam,
                metrics=['acc'])

history = Kim_G.fit(X_train, y_train, validation_split=0.33, epochs=9, batch_size=600)
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(loss) + 1)
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

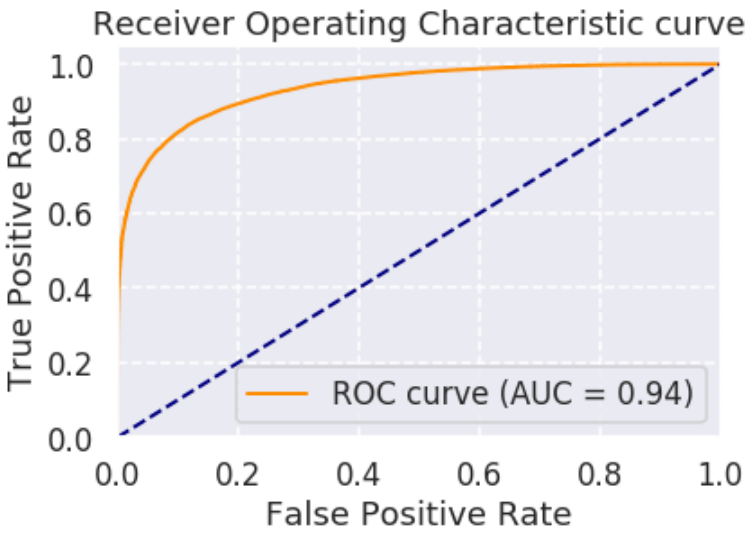
```
rain on 81492 samples, validate on 40138 samples
poch 1/9
1492/81492 [==============================] - 55s 681us/sample - loss: 0.4692 - acc: 0.7662
poch 2/9
1492/81492 [==============================] - 55s 671us/sample - loss: 0.3602 - acc: 0.8351
poch 3/9
1492/81492 [==============================] - 55s 670us/sample - loss: 0.3298 - acc: 0.8536
poch 4/9
1492/81492 [==============================] - 55s 678us/sample - loss: 0.3083 - acc: 0.8639
poch 5/9
1492/81492 [==============================] - 55s 680us/sample - loss: 0.2932 - acc: 0.8726
poch 6/9
1492/81492 [==============================] - 56s 687us/sample - loss: 0.2789 - acc: 0.8803
poch 7/9
1492/81492 [==============================] - 55s 680us/sample - loss: 0.2648 - acc: 0.8892
poch 8/9
1492/81492 [==============================] - 56s 686us/sample - loss: 0.2527 - acc: 0.8946
poch 9/9
1492/81492 [==============================] - 56s 682us/sample - loss: 0.2394 - acc: 0.9020
```




```
# Calculate outputs in test set
prob_test_GK = Kim_G.predict(X_test, verbose = 1)
prob_train = Kim_G.predict(X_train, verbose = 1)

# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(y_train, prob_train)
roc_auc = auc(fpr, tpr)
```

```
print('\nAUC train: ', roc_auc)

# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(y_test, prob_test_GK)
roc_auc = auc(fpr, tpr)
print('AUC test: ', roc_auc)

sns.set('talk', 'darkgrid', 'dark', font_scale=1, \
        rc={"lines.linewidth": 2, 'grid.linestyle': '--'})

lw = 2
plt.figure()
plt.plot(fpr, tpr, color='darkorange',
         lw=lw, label='ROC curve (AUC = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic curve')
plt.legend(loc="lower right")
plt.savefig('roc_auc.pdf', format = "pdf")
plt.show()
```

```
59908/59908 [==============================] - 16s 262us/sample
121630/121630 [==============================] - 31s 259us/sample

AUC train:  0.9610499658618654
AUC test:  0.9371429013926638
```

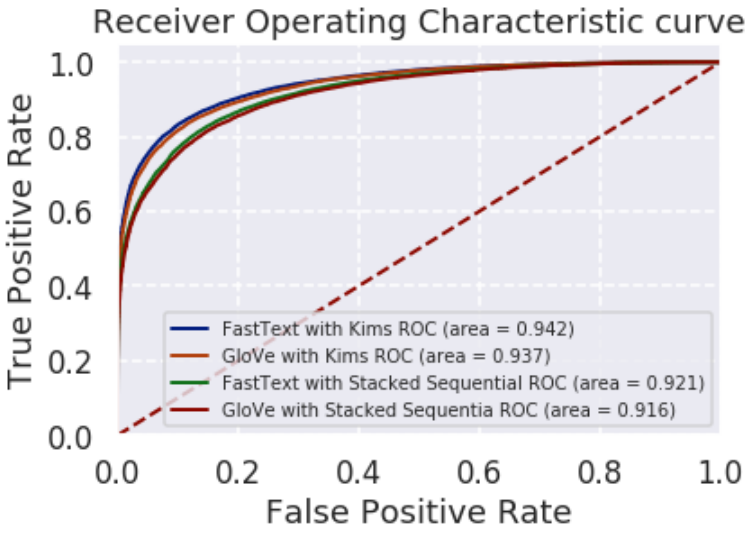



```
prob_test_GK[prob_test_GK > Cutoff] = 1
prob_test_GK[prob_test_GK <= Cutoff] = 0


confusion_matrix4 = \
confusion_matrix(y_true = y_test, y_pred = prob_test_GK)


confusion_matrix4
```

```
array([[15656,  3942],
       [ 4270, 36040]])
```

Performance

```
models = [
{
    'label': 'FastText with Kims',
    'probs': prob_test_FK
},
{
    'label': 'GloVe with Kims',
    'probs': prob_test_GK
},
{
    'label': 'FastText with Stacked Sequential',
    'probs': prob_test_FS
},
{
    'label': 'GloVe with Stacked Sequentia',
    'probs': prob_test_GS
}
]


for m in models:

  fpr, tpr, thresholds = roc_curve(y_test.astype('float'),
                                   m['probs'])
  roc_auc = auc(fpr, tpr)
  plt.plot(fpr, tpr, label='%s ROC (area = %0.3f)' % (m['label'], roc_auc))

lw = 2
# Settings
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic curve')
plt.legend(loc="lower right", prop={"size":10})

# Plot!
plt.show()
```


Receiver Operating Characteristic curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
FastText with Kims ROC (area = 0.942)
GloVe with Kims ROC (area = 0.937)
FastText with Stacked Sequential ROC (area = 0.921)
GloVe with Stacked Sequentia ROC (area = 0.916)