In [ ]:

```
import shutil
import os
import pandas as pd
import re

import nltk
nltk.download('maxent_ne_chunker')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('words')

from nltk import ne_chunk, pos_tag, word_tokenize
from nltk.tree import Tree

from keras.models import Sequential
from keras.layers import Dense
from keras.layers.embeddings import Embedding
from keras.layers import Dense, Activation
from tensorflow.keras import initializers
from sklearn.metrics import f1_score
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.callbacks import LearningRateScheduler
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.text import one_hot
from keras.preprocessing import sequence
```

```
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Package words is already up-to-date!
```

In [ ]:

```
files = os.listdir('documents')

sentence = []
filenames2 = []
for file in files:
  filenames2.append(file)
  if(file.endswith("txt")):
    data = open('documents/'+file,'rb').read()
    sentence.append(data)

sub_file_name = []
for i in filenames2:
  count = 0
  for j in i.strip(''):
    count += 1
    if '_' == j:
      sub_file_name.append(i[0:count-1])
      break

unique = list(set(sub_file_name))

dic = {}
for i in range(len(unique)):
  dic[unique[i]] = i

label = []
for i in sub_file_name:
  label.append(dic[i])

data = {'ID':filenames2 ,'sentence':sentence, 'label':label}
data_df = pd.DataFrame(data)
data_df.to_pickle('cnn_text')
```

In [ ]:

```
data = pd.read_pickle('cnn_text')
```

In [ ]:

```
# for email
emails = []
email_list = []
for z in range(len(data['sentence'])):
  email = re.findall(r"[a-zA-Z0-9\.\-+_]+@[a-zA-Z0-9\.\-+_]+\.[a-zA-Z]+", str(data['sentence'][z])) #source from tutorial point
  a = []
  for i in email:
    a.append(i)
  emails.append(a)
  filter_email = []
  for m in a:
    loc = 0
    for n, k in enumerate(m):
      if '@' == k:
        loc = n
        break
    ema = m[loc+1:]
    for o in ema.split("."):
      if o != 'com' and len(o) > 2:
        filter_email.append(o)
  email_list.append(' '.join(filter_email))
print(email_list[1])
```

```
mantis mantis mantis
```

In [ ]:

```
data['sentence'][4]
```

Out[ ]:

```
b'From: strom@Watson.Ibm.Com (Rob Strom)\nSubject: Re: [soc.motss, et al.] "Princeton axes matching funds for Boy Scouts"\n\nIn article
<N4HY.93Apr5120934@harder.ccr-p.ida.org>, n4hy@harder.ccr-p.ida.org (Bob McGwier) writes:\n\n|> [1] HOWEVER, I hate economic terrorism a
nd political correctness\n|> worse than I hate this policy.  \n\n\n|> [2] A more effective approach is to stop donating\n|> to ANY organ
izating that directly or indirectly supports gay rights issues\n|> until they end the boycott on funding of scouts.  \n\nCan somebody re
concile the apparent contradiction between [1] and [2]?\n\n-- \nRob Strom, strom@watson.ibm.com, (914) 784-7641\nIBM Research, 30 Saw Mi
ll River Road, P.O. Box 704, Yorktown Heights, NY  10598\n'
```

In [ ]:

```
#for subject
subjects = []
for i in range(len(data['sentence'])):
  li = []
  a = [j for j in data['sentence'][i].splitlines( )]
  li.append(a[1])

  ind = []
  for k , l in enumerate(str(li[0]).strip('')):
    if ':' == l:
      ind.append(k+1)

  z=str(li[0])[ind[len(ind)-1]:]
  special_characters = "'""!@#$%^&*()-+?_=,<>/""'" # source from tutorial point
  f = []
  s = ''
  for j , k in enumerate(z.strip()):
    if k  not in special_characters:
      f.append(k)
  subjects.append(s.join(f))
```

In [ ]:

```
#filtered sentence
filter_sentence = []
for i in range(len(data['sentence'])):
  d = str(data['sentence'][i])
  a = [j for j in data['sentence'][i].splitlines( )]
  c = a[1]
  h = str(c)[2:-1]

  new_string = d.replace(h, "")
  for k in range(len(emails[i])):
    new_string = new_string.replace(emails[i][k], "")
  filter_sentence.append(new_string)

print(filter_sentence[4])
```

```
b'From:  (Rob Strom)\n\n\nIn article <>,  (Bob McGwier) writes:\n\n|> [1] HOWEVER, I hate economic terrorism and political correctness\n
|> worse than I hate this policy.  \n\n\n|> [2] A more effective approach is to stop donating\n|> to ANY organizating that directly or i
ndirectly supports gay rights issues\n|> until they end the boycott on funding of scouts.  \n\nCan somebody reconcile the apparent contr
adiction between [1] and [2]?\n\n-- \nRob Strom, , (914) 784-7641\nIBM Research, 30 Saw Mill River Road, P.O. Box 704, Yorktown Heights,
NY  10598\n'
```

In [ ]:

```
data2 = {'text':data['sentence'],'class':sub_file_name,'filtered_text':filter_sentence,'preprocessed_subject':subjects,
         'preprocessed_emails':email_list,'label':data['label']}
df = pd.DataFrame(data2)
df.to_pickle('full_data')
```

In [ ]:

```
df = pd.read_pickle('full_data')
df
```

Out[ ]:

	text	class	filtered_text	preprocessed_subject	preprocessed_emails	label
0	b'From: mathew <mathew@mantis.co.uk>\nSubject:...	alt.atheism	b'From: mathew <>\n\n\nArchive-name: atheism/r...	Atheist Resources	mantis netcom mantis	3
1	b'From: mathew <mathew@mantis.co.uk>\nSubject:...	alt.atheism	b'From: mathew <>\n\n\nArchive-name: atheism/i...	Introduction to Atheism	mantis mantis mantis	3
2	b'From: I3150101@dbstu1.rz.tu-bs.de (Benedikt ...	alt.atheism	b'From: (Benedikt Rosenau)\n\nIn article <>...	Gospel Dating	dbstu1 tu-bs mimsy umd edu umd edu	3
3	b'From: mathew <mathew@mantis.co.uk>\nSubject:...	alt.atheism	b'From: mathew <>\n\n\ (...until kings become ...	university violating separation of churchstate	mantis kepler unh edu	3
4	b'From: strom@Watson.Ibm.Com (Rob Strom)\nSubj...	alt.atheism	b'From: (Rob Strom)\n\n\nIn article <>, (Bob...	[soc.motss et al.] "Princeton axes matching fu...	Watson Ibm Com harder ccr-p ida org harder ccr...	3
...	...	...	...	...	...	...

	text	class	filtered_text	preprocessed_subject	preprocessed_emails	label
18823	b'From: sbuckley@fraser.sfu.ca (Stephen Buckle...	talk.religion.misc	b'From: (Stephen Buckley)\n\n\ (Paul D Boxrud...	Religion and marriage	fraser sfu magnus acs ohio-state edu	4
18824	b'From: bakerj@gtephx.UUCP (Jon Baker)\nSubjec...	talk.religion.misc	b'From: (Jon Baker)\n\n\nIn article <>, (Joa...	How do you know what happened	gtephx UUCP ifi uio ifi uio ncratl AtlantaGA N...	4
18825	b"From: pharvey@quack.kfu.com (Paul Harvey)\nS...	talk.religion.misc	b"From: (Paul Harvey)\n\n\nIn article <> \ (B...	Why did they behave as they did Wacoreading su...	quack kfu emx utexas edu emx utexas edu	4
18826	b'From: <KEVXU@CUNYVM.BITNET>\nSubject: Re: In...	talk.religion.misc	b'From: <>\n\n\n\nIn article <>\ (Gerry Palo)\...	Info about New Age	CUNYVM BITNET digi lonestar org digi lonestar org	4
18827	b"From: pharvey@quack.kfu.com (Paul Harvey)\nS...	talk.religion.misc	b"From: (Paul Harvey)\n\n\nIn article <> \ (B...	After 2000 years etc"	quack kfu darkside osrhe uoknor edu okcforum o...	4

18828 rows × 6 columns

In [ ]:

```
df['filtered_text'][0]
```

Out[ ]:

```
'b\'From: mathew <>\\n\\n\\nArchive-name: atheism/resources\\nAlt-atheism-archive-name: resources\\nLast-modified: 11 December 1992\\nVe
rsion: 1.0\\n\\n                                   Atheist Resources\\n\\n                          Addresses of Atheist Organizations\\n\\n
USA\\n\\nFREEDOM FROM RELIGION FOUNDATION\\n\\nDarwin fish bumper stickers and assorted other atheist paraphernalia are\\navailable from
the Freedom From Religion Foundation in the US.\\n\\nWrite to:  FFRF, P.O. Box 750, Madison, WI 53701.\\nTelephone: (608) 256-8900\\n\\n
EVOLUTION DESIGNS\\n\\nEvolution Designs sell the "Darwin fish".  It\\\'s a fish symbol, like the ones\\nChristians stick on their cars,
but with feet and the word "Darwin" written\\ninside.  The deluxe moulded 3D plastic fish is $4.95 postpaid in the US.\\n\\nWrite to:  E
volution Designs, 7119 Laurel Canyon #4, North Hollywood,\\n           CA 91605.\\n\\nPeople in the San Francisco Bay area can get Darwi
n Fish from Lynn Gold --\\ntry mailing <>.  For net people who go to Lynn directly, the\\nprice is $4.95 per fish.\\n\\nAMERICAN ATHEIST
PRESS\\n\\nAAP publish various atheist books -- critiques of the Bible, lists of\\nBiblical contradictions, and so on.  One such book i
s:\\n\\n"The Bible Handbook" by W.P. Ball and G.W. Foote.  American Atheist Press.\\n372 pp.  ISBN 0-910309-26-4, 2nd edition, 1986.  Bi
ble contradictions,\\nabsurdities, atrocities, immoralities... contains Ball, Foote: "The Bible\\nContradicts Itself", AAP.  Based on th
e King James version of the Bible.\\n\\nWrite to:  American Atheist Press, P.O. Box 140195, Austin, TX 78714-0195.\\n      or:  7215 Cam
eron Road, Austin, TX 78752-2973.\\nTelephone: (512) 458-1244\\nFax:       (512) 467-9525\\n\\nPROMETHEUS BOOKS\\n\\nSell books includin
g Haught\\\'s "Holy Horrors" (see below).\\n\\nWrite to:  700 East Amherst Street, Buffalo, New York 14215.\\nTelephone: (716) 837-247
5.\\n\\nAn alternate address (which may be newer or older) is:\\nPrometheus Books, 59 Glenn Drive, Buffalo, NY 14228-2197.\\n\\nAFRICAN-
AMERICANS FOR HUMANISM\\n\\nAn organization promoting black secular humanism and uncovering the history of\\nblack freethought.  They pu
blish a quarterly newsletter, AAH EXAMINER.\\n\\nWrite to:  Norm R. Allen, Jr., African Americans for Humanism, P.O. Box 664,\\n
Buffalo, NY 14226.\\n\\n                                United Kingdom\\n\\nRationalist Press Association          National Secular Soci
ety\\n88 Islington High Street               702 Holloway Road\\nLondon N1 8EW                                London N19 3NL\\n071 226 7251
071 272 1266\\n\\nBritish Humanist Association           South Place Ethical Society\\n14 Lamb\\\'s Conduit Passage              Conway
Hall\\nLondon WC1R 4RH                       Red Lion Square\\n071 430 0908                              London WC1R 4RL\\nfax 071 430 127
1                    071 831 7723\\n\\nThe National Secular Society publish "The Freethinker", a monthly magazine\\nfounded in 188
1.\\n\\n                                 Germany\\n\\nIBKA e.V.\\nInternationaler Bund der Konfessionslosen und Atheisten\\nPostfach 8
80, D-1000 Berlin 41. Germany.\\n\\nIBKA publish a journal:\\nMIZ. (Materialien und Informationen zur Zeit. Politisches\\nJournal der Ko
nfessionslosesn und Atheisten. Hrsg. IBKA e.V.)\\nMIZ-Vertrieb, Postfach 880, D-1000 Berlin 41. Germany.\\n\\nFor atheist books, write t
o:\\n\\nIBDK, Internationaler B"ucherdienst der Konfessionslosen\\nPostfach 3005, D-3000 Hannover 1. Germany.\\nTelephone: 0511/211216
\\n\\n\\n                               Books -- Fiction\\n\\nTHOMAS M. DISCH\\n\\n"The Santa Claus Compromise"\\nShort story.  The ulti
mate proof that Santa exists.  All characters and \\nevents are fictitious.  Any similarity to living or dead gods -- uh, well...\\n\\nW
ALTER M. MILLER, JR\\n\\n"A Canticle for Leibowitz"\\nOne gem in this post atomic doomsday novel is the monks who spent their lives\\nco
pying blueprints from "Saint Leibowitz", filling the sheets of paper with\\nink and leaving white lines and letters.\\n\\nEDGAR PANGBORN
\\n\\n"Davy"\\nPost atomic doomsday novel set in clerical states.  The church, for example,\\nforbids that anyone "produce, describe or
use any substance containing...\\natoms". \\n\\nPHILIP K. DICK\\n\\nPhilip K. Dick Dick wrote many philosophical and thought-provoking s
hort \\nstories and novels.  His stories are bizarre at times, but very approachable.\\nHe wrote mainly SF, but he wrote about people, t
ruth and religion rather than\\ntechnology.  Although he often believed that he had met some sort of God, he\\nremained sceptical.  Amon
gst his novels, the following are of some relevance:\\n\\n"Galactic Pot-Healer"\\nA fallible alien deity summons a group of Earth crafts
men and women to a\\nremote planet to raise a giant cathedral from beneath the oceans.  When the\\ndeity begins to demand faith from the
earthers, pot-healer Joe Fernwright is\\nunable to comply.  A polished, ironic and amusing novel.\\n\\n"A Maze of Death"\\nNoteworthy fo
r its description of a technology-based religion.\\n\\n"VALIS"\\nThe schizophrenic hero searches for the hidden mysteries of Gnostic\\nC
hristianity after reality is fired into his brain by a pink laser beam of\\nunknown but possibly divine origin.  He is accompanied by hi
s dogmatic and\\ndismissively atheist friend and assorted other odd characters.\\n\\n"The Divine Invasion"\\nGod invades Earth by making
a young woman pregnant as she returns from\\nanother star system.  Unfortunately she is terminally ill, and must be\\nassisted by a dead
man whose brain is wired to 24-hour easy listening music.\\n\\nMARGARET ATWOOD\\n\\n"The Handmaid\\\'s Tale"\\nA story based on the prem
ise that the US Congress is mysteriously\\nassassinated, and fundamentalists quickly take charge of the nation to set it\\n"right" agai
n.  The book is the diary of a woman\\\'s life as she tries to live\\nunder the new Christian theocracy.  Women\\\'s right to own proper
ty is revoked,\\nand their bank accounts are closed; sinful luxuries are outlawed, and the\\nradio is only used for readings from the Bi
ble.  Crimes are punished\\nretroactively: doctors who performed legal abortions in the "old world" are\\nhunted down and hanged.  Atwoo
d\\\'s writing style is difficult to get used to\\nat first, but the tale grows more and more chilling as it goes on.\\n\\nVARIOUS AUTHO
RS\\n\\n"The Bible"\\nThis somewhat dull and rambling work has often been criticized.  However, it\\nis probably worth reading, if only
so that you\\\'ll know what all the fuss is\\nabout.  It exists in many different versions, so make sure you get the one\\ntrue versio
n.\\n\\n                             Books -- Non-fiction\\n\\nPETER DE ROSA\\n\\n"Vicars of Christ", Bantam Press, 1988\\nAlthough de R
osa seems to be Christian or even Catholic this is a very\\nenlighting history of papal immoralities, adulteries, fallacies etc.\\n(Germ
an translation: "Gottes erste Diener. Die dunkle Seite des Papsttums",\\nDroemer-Knaur, 1989)\\n\\nMICHAEL MARTIN\\n\\n"Atheism: A Philo
sophical Justification", Temple University Press,\\n Philadelphia, USA.\\nA detailed and scholarly justification of atheism.  Contains a
n outstanding\\nappendix defining terminology and usage in this (necessarily) tendentious\\narea.  Argues both for "negative atheism"
(i.e. the "non-belief in the\\nexistence of god(s)") and also for "positive atheism" ("the belief in the\\nnon-existence of god(s)").  I
ncludes great refutations of the most\\nchallenging arguments for god; particular attention is paid to refuting\\ncontempory theists suc
h as Platinga and Swinburne.\\n541 pages. ISBN 0-87722-642-3 (hardcover; paperback also available)\\n\\n"The Case Against Christianity",
Temple University Press\\nA comprehensive critique of Christianity, in which he considers\\nthe best contemporary defences of Christiani
ty and (ultimately)\\ndemonstrates that they are unsupportable and/or incoherent.\\n273 pages. ISBN 0-87722-767-5\\n\\nJAMES TURNER\\n
\\n"Without God, Without Creed", The Johns Hopkins University Press, Baltimore,\\n MD, USA\\nSubtitled "The Origins of Unbelief in Ameri
ca".  Examines the way in which\\nunbelief (whether agnostic or atheistic)  became a mainstream alternative\\nworld-view.  Focusses on t
he period 1770-1900, and while considering France\\nand Britain the emphasis is on American, and particularly New England\\ndevelopment
s.  "Neither a religious history of secularization or atheism,\\nWithout God, Without Creed is, rather, the intellectual history of the
fate\\nof a single idea, the belief that God exists." \\n316 pages. ISBN (hardcover) 0-8018-2494-X (paper) 0-8018-3407-4\\n\\nGEORGE SEL
DES (Editor)\\n\\n"The great thoughts", Ballantine Books, New York, USA\\nA "dictionary of quotations" of a different kind, concentratin
g on statements\\nand writings which, explicitly or implicitly, present the person\\\'s philosophy\\nand world-view.  Includes obscure
(and often suppressed) opinions from many\\npeople.  For some popular observations, traces the way in which various\\npeople expressed a
nd twisted the idea over the centuries.  Quite a number of\\nthe quotations are derived from Cardiff\\\'s "What Great Men Think of Relig
ion"\\nand Noyes\\\' "Views of Religion".\\n490 pages. ISBN (paper) 0-345-29887-X.\\n\\nRICHARD SWINBURNE\\n\\n"The Existence of God (Re
vised Edition)", Clarendon Paperbacks, Oxford\\nThis book is the second volume in a trilogy that began with "The Coherence of\\nTheism"
(1977) and was concluded with "Faith and Reason" (1981).  In this\\nwork, Swinburne attempts to construct a series of inductive argument
s for the\\nexistence of God.  His arguments, which are somewhat tendentious and rely\\nupon the imputation of late 20th century western
Christian values and\\naesthetics to a God which is supposedly as simple as can be conceived, were\\ndecisively rejected in Mackie\\\'s
```

```
"The Miracle of Theism".  In the revised\\nedition of "The Existence of God", Swinburne includes an Appendix in which he\\nmakes a somew
hat incoherent attempt to rebut Mackie.\\n\\nJ. L. MACKIE\\n\\n"The Miracle of Theism", Oxford\\nThis (posthumous) volume contains a com
prehensive review of the principal\\narguments for and against the existence of God.  It ranges from the classical\\nphilosophical posit
ions of Descartes, Anselm, Berkeley, Hume et al, through\\nthe moral arguments of Newman, Kant and Sidgwick, to the recent restatements 
\\nof the classical theses by Plantinga and Swinburne.  It also addresses those\\npositions which push the concept of God beyond the rea
lm of the rational,\\nsuch as those of Kierkegaard, Kung and Philips, as well as "replacements for\\nGod" such as Lelie\\\'s axiarchism.
The book is a delight to read - less\\nformalistic and better written than Martin\\\'s works, and refreshingly direct\\nwhen compared wi
th the hand-waving of Swinburne.\\n\\nJAMES A. HAUGHT\\n\\n"Holy Horrors: An Illustrated History of Religious Murder and Madness",\\n Pr
ometheus Books\\nLooks at religious persecution from ancient times to the present day -- and\\nnot only by Christians.\\nLibrary of Cong
ress Catalog Card Number 89-64079. 1990.\\n\\nNORM R. ALLEN, JR.\\n\\n"African American Humanism: an Anthology"\\nSee the listing for Af
rican Americans for Humanism above.\\n\\nGORDON STEIN\\n\\n"An Anthology of Atheism and Rationalism", Prometheus Books\\nAn anthology co
vering a wide range of subjects, including \\\'The Devil, Evil\\nand Morality\\\' and \\\'The History of Freethought\\\'.  Comprehensive
bibliography.\\n\\nEDMUND D. COHEN\\n\\n"The Mind of The Bible-Believer", Prometheus Books\\nA study of why people become Christian fund
amentalists, and what effect it\\nhas on them.\\n\\n                              Net Resources\\n\\nThere\\\'s a small mail-based arc
hive server at mantis.co.uk which carries\\narchives of old alt.atheism.moderated articles and assorted other files.  For\\nmore informa
tion, send mail to  saying\\n\\n   help\\n\\n   send atheism/index\\n\\nand it will mail back a reply.\\n\\n\\nmathew\\n\\xff\\n\''
```

In [ ]:

```python
# https://stackoverflow.com/a/47091490/4084039
import re

def decontracted(phrase):
    # specific
    phrase = re.sub(r"won't", "will not", phrase)
    phrase = re.sub(r"can\'t", "can not", phrase)

    # general
    phrase = re.sub(r"n\'t", " not", phrase)
    phrase = re.sub(r"\'re", " are", phrase)
    phrase = re.sub(r"\'s", " is", phrase)
    phrase = re.sub(r"\'d", " would", phrase)
    phrase = re.sub(r"\'ll", " will", phrase)
    phrase = re.sub(r"\'t", " not", phrase)
    phrase = re.sub(r"\'ve", " have", phrase)
    phrase = re.sub(r"\'m", " am", phrase)
    return phrase

#https://stackoverflow.com/questions/31836058/nltk-named-entity-recognition-to-a-python-list/31837224#31837224
def chunk(string):
  chunked = ne_chunk(pos_tag(word_tokenize(string)))
  current_chunk = []
  for i in chunked:
    if type(i) == Tree:
      c = str(i)
      if "PERSON" not in c:
        complete = "_".join([token for token, pos in i.leaves()])
        f = []
        for j in complete.split('_'):
          if len(j) > 2:
            f.append(j)
        g = '_'.join(f)
        current_chunk.append(g)

    else:
      current_chunk.append(i[0])
  chu = ' '.join(current_chunk)

  return chu

def remove_underscore(string):
  e = []
  for i in string.split(' '):
    if len(i) >=1:
      if i[0] == '_' and i[len(i)-1] == '_':
        e.append(i[1:(len(i)-1)])
        continue
      if i[0] == '_':
        e.append(i[1:])
        continue
      if i[len(i)-1] == '_':
        e.append(i[0:(len(i)-1)])
        continue
      else:
        e.append(i)

  return ' '.join(e)
```

In [ ]:

```python
def preprocess(Input_Text):
  preprocessed_string = []
  for i in range(len(Input_Text)):
    x = Input_Text[i].replace("from:", "").replace("write to:", "").replace('\\n',' ').replace('\\t',' ')
    remove_char_word = lambda y: ' '.join(j for j in y.split() if ':' not in j)
    x = remove_char_word(x)
    #https://stackoverflow.com/questions/14596884/remove-text-between-and-in-python/14598135
    x = re.sub("[\(\[].*?[\)\]]", "", x)
    x = decontracted(x)
    #remove spacial character: https://stackoverflow.com/a/5843547/4084039
    x = re.sub('[^A-Za-z0-9]+', ' ', x)
    x = chunk(x)
    #to remove digits
    x = re.sub('[0-9]+', '', x)
    x = remove_underscore(x)
    x = x.lower()
    #word length
    h = []
    for i in x.split(' '):
      if len(i) <15 and len(i) > 2:
        h.append(i)
      else:
        continue
    x = ' '.join(h)
    preprocessed_string.append(x)

  return  preprocessed_string
```

In [ ]:
```
preprocessed_text = preprocess(df['filtered_text'])
```

In [ ]:
```
preprocessed_text[5000]
```

Out[ ]:
```
'some one asked recently why they when they used see they could create given size seemed imply they could but the server did not create
cursors that size investigation showed that some servers will happily return any size the size the root window while others return some
fixed limit more reasonable size the interesting thing that the same server binary acts differently different hardware sun with will cla
im cursors root window size are while sun with will stop far have also seen this behavior ncd and terminals and have been told also occu
rs hps actually the ncd even more liberal sizes much larger then the root winodw are gladly returned semi broken this behavior correct w
ould really like see cursor med sitt skjegg research lokkar borni under sole vegg box boulder gjo med sitt shinn jagar borni inn'
```

In [ ]:
```
preprocessed_subject = preprocess(df['preprocessed_subject'])
```

In [ ]:
```
preprocessed_email = preprocess(df['preprocessed_emails'])
```

In [ ]:
```
data3 = {'text':data['sentence'],'class':sub_file_name,'preprocessed_text':preprocessed_text,'preprocessed_subject':preprocessed_subject,
         'preprocessed_email':preprocessed_email,'label':data['label']}
df = pd.DataFrame(data3)
df.to_pickle('preprocessed_data')
```

In [ ]:
```
df = pd.read_pickle('preprocessed_data')
df
```

Out[ ]:

	text	class	preprocessed_text	preprocessed_subject	preprocessed_email	label
0	b'From: mathew <mathew@mantis.co.uk>\nSubject:...	alt.atheism	mathew atheism resources resources december us...	atheist resources	mantis netcom mantis	3
1	b'From: mathew <mathew@mantis.co.uk>\nSubject:...	alt.atheism	mathew atheism introduction introduction april...	introduction atheism	mantis mantis mantis	3
2	b'From: I3150101@dbstu1.rz.tu-bs.de (Benedikt ...	alt.atheism	article has quite different not necessarily mo...	dating	dbstu mimsy umd edu umd edu	3
3	b'From: mathew <mathew@mantis.co.uk>\nSubject:...	alt.atheism	mathew recently ras have been ordered post rel...	university violating separation churchstate	mantis kepler unh edu	3
4	b'From: strom@Watson.Ibm.Com (Rob Strom)\nSubj...	alt.atheism	article however hate economic terrorism and po...	princeton axes matching funds for	ibm_com harder ccr ida org harder ccr ida org ...	3
...	...	...	...	...	...	...
18823	b'From: sbuckley@fraser.sfu.ca (Stephen Buckle...	talk.religion.misc	wasn not sure this was the right newsgroup pos...	religion and marriage	fraser sfu magnus acs ohio state edu	4
18824	b'From: bakerj@gtephx.UUCP (Jon Baker)\nSubjec...	talk.religion.misc	article article probably not but then don not ...	how you know what happened	gtephx uucp ifi uio ifi uio ncratl atlantaga n...	4
18825	b"From: pharvey@quack.kfu.com (Paul Harvey)\nS...	talk.religion.misc	article you would like understand better the s...	why did they behave they did wacoreading sugge...	quack kfu emx utexas edu emx utexas edu	4
18826	b'From: <KEVXU@CUNYVM.BITNET>\nSubject: Re: In...	talk.religion.misc	article the danger anti cult groups that while...	info about new age	cunyvm bitnet digi lonestar org digi lonestar org	4
18827	b"From: pharvey@quack.kfu.com (Paul Harvey)\nS...	talk.religion.misc	article once you enter here your terminal beom...	after years etc	quack kfu darkside osrhe uoknor edu okcforum o...	4

18828 rows × 6 columns

In [ ]:
```
#concatinating preprocessed_text, preprocessed_subject and preprocessed_email
X = df['preprocessed_text'] +' '+ df['preprocessed_subject'] +' '+ df['preprocessed_email']
Y = df['label']
```

In [ ]:
```
# train test split
from sklearn.model_selection import train_test_split
import numpy as np
index = index = np.arange(len(X))
X_train, X_test, y_train, y_test, i_train, i_test = train_test_split(X, Y,index, test_size=0.25, stratify=Y)
```

In [ ]:
```
len(X_train) , len(X_test)
```

Out[ ]:
```
(14121, 4707)
```

In [ ]:
```
li = []
for i in X_train:
  li.append(len(i.split(' ')))

max_length = max(li)
print(max_length)
```

```
8806
```

In [ ]:

```
#https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/
from keras.preprocessing.text import Tokenizer
t = Tokenizer(filters='!"#$%&()*+,-./:;<=>?@[\\]^`{|}~\t\n')
t.fit_on_texts(X_train)
vocab_size = len(t.word_index) + 1
# integer encode the documents
X_train1 = t.texts_to_sequences(X_train)

X_test1 = t.texts_to_sequences(X_test)
max_length = max_length
X_train2 = sequence.pad_sequences(X_train1, maxlen=max_length, padding='post')
X_test2 = sequence.pad_sequences(X_test1, maxlen=max_length, padding='post')

import pickle
with open('glove_vectors', 'rb') as f:
 model = pickle.load(f)
 embeddings_index = dict(zip(model.keys(),model.values()))
 f.close()

print('Loaded %s word vectors.' % len(embeddings_index))
#create a weight matrix for words in training docs
#each word is of 300 dimension
embedding_matrix = np.zeros((vocab_size, 300))
for word, i in t.word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector
print('Loaded %s word vectors.' % len(embedding_matrix))
```

```
Loaded 51510 word vectors.
Loaded 75967 word vectors.
```

In [ ]:

```
sen_length = []
for i in X_train:
  sen_length.append(len(i.split()))
max_total_length = max(sen_length)
print(max_total_length)
```

```
8806
```

In [ ]:

```
from sklearn import preprocessing
y_train = np.array(y_train)
y_test = np.array(y_test)
label = preprocessing.LabelEncoder()
y_train = label.fit_transform(y_train)
y_test = label.fit_transform(y_test)
y_train = tf.keras.utils.to_categorical(y_train)
y_test = tf.keras.utils.to_categorical(y_test)
```

In [ ]:

```
from tensorflow.keras.layers import Dense,Input,Conv1D,MaxPooling1D,Activation,Dropout,Flatten
from tensorflow.keras.models import Model
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers.embeddings import Embedding
from keras.preprocessing import sequence
import random as rn
import keras
import os
os.environ['PYTHONHASHSEED'] = '0'

##https://keras.io/getting-started/faq/#how-can-i-obtain-reproducible-results-using-keras-during-development
tf.keras.backend.clear_session()

## Set the random seed values to regenerate the model.
np.random.seed(0)
rn.seed(0)

embedding_vecor_length = 30

inputs = Input(shape=(max_review_length,), dtype='int32', name='inputs')

essay_embedding = Embedding(vocab_size, 300, weights=[embedding_matrix], input_length=max_length, trainable=True)(inputs)

con1 = Conv1D(filters=16, kernel_size=16, activation='relu')(essay_embedding)
con2 = Conv1D(filters=16, kernel_size=16, activation='relu')(essay_embedding)
con3 = Conv1D(filters=16, kernel_size=16, activation='relu')(essay_embedding)

combine_con1 = keras.layers.concatenate([con1, con2,con3])

max_pool1 = MaxPooling1D(pool_size=2)(combine_con1)

con4 = Conv1D(filters=16, kernel_size=16, activation='relu')(max_pool1)
con5 = Conv1D(filters=16, kernel_size=16, activation='relu')(max_pool1)
con6 = Conv1D(filters=16, kernel_size=16, activation='relu')(max_pool1)
```

```
combine_con2 = keras.layers.concatenate([con4, con5,con6])

max_pool2 = MaxPooling1D(pool_size=2)(combine_con2)

con7 = Conv1D(filters=16, kernel_size=16, activation='relu')(max_pool2)

flatten = Flatten()(con7)

drop_out = Dropout(0.2)(flatten)

dense1 = Dense(40, activation='relu')(drop_out)


Out = Dense(units=20,activation='softmax')(dense1)

#Creating a model
model = Model(inputs=inputs,outputs=Out)
```

In [ ]:

```
model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to                     
==================================================================================================
inputs (InputLayer)             [(None, 8806)]       0                                            
__________________________________________________________________________________________________
embedding (Embedding)           (None, 8806, 300)    22790100    inputs[0][0]                     
__________________________________________________________________________________________________
conv1d (Conv1D)                 (None, 8791, 16)     76816       embedding[0][0]                  
__________________________________________________________________________________________________
conv1d_1 (Conv1D)               (None, 8791, 16)     76816       embedding[0][0]                  
__________________________________________________________________________________________________
conv1d_2 (Conv1D)               (None, 8791, 16)     76816       embedding[0][0]                  
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 8791, 48)     0           conv1d[0][0]                     
                                                                 conv1d_1[0][0]                   
                                                                 conv1d_2[0][0]                   
__________________________________________________________________________________________________
max_pooling1d (MaxPooling1D)    (None, 4395, 48)     0           concatenate[0][0]                
__________________________________________________________________________________________________
conv1d_3 (Conv1D)               (None, 4380, 16)     12304       max_pooling1d[0][0]              
__________________________________________________________________________________________________
conv1d_4 (Conv1D)               (None, 4380, 16)     12304       max_pooling1d[0][0]              
__________________________________________________________________________________________________
conv1d_5 (Conv1D)               (None, 4380, 16)     12304       max_pooling1d[0][0]              
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 4380, 48)     0           conv1d_3[0][0]                   
                                                                 conv1d_4[0][0]                   
                                                                 conv1d_5[0][0]                   
__________________________________________________________________________________________________
max_pooling1d_1 (MaxPooling1D)  (None, 2190, 48)     0           concatenate_1[0][0]              
__________________________________________________________________________________________________
conv1d_6 (Conv1D)               (None, 2175, 16)     12304       max_pooling1d_1[0][0]            
__________________________________________________________________________________________________
flatten (Flatten)               (None, 34800)        0           conv1d_6[0][0]                   
__________________________________________________________________________________________________
dropout (Dropout)               (None, 34800)        0           flatten[0][0]                    
__________________________________________________________________________________________________
dense (Dense)                   (None, 40)           1392040     dropout[0][0]                    
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 20)           820         dense[0][0]                      
==================================================================================================
Total params: 24,462,624
Trainable params: 24,462,624
Non-trainable params: 0
__________________________________________________________________________________________________
```

In [ ]:

```
optimizer = tf.keras.optimizers.Adam(
    learning_rate=0.001,
    beta_1=0.9,
    beta_2=0.999,
    epsilon=1e-07,
    amsgrad=False,
    name="Adam",

)
```

In [ ]:

```
#compiling
model.compile(optimizer=optimizer,loss='categorical_crossentropy',metrics=['accuracy'])
```

In [ ]:

```
from sklearn.metrics import f1_score
from sklearn.metrics import roc_auc_score
class f1_score_and_auc_Callback(tf.keras.callbacks.Callback):

    def  on_train_begin(self,logs={}):
      self.f1_micro=[]
      self.auc_score=[]

    def on_epoch_end(self, epoch, logs=None):
      y_pred=self.model.predict(X_test2).round()
```

```
        y_pred_=self.model.predict(X_test2)
        y_true=y_test
        score=f1_score(y_true, y_pred, average='samples')
        Auc_score  = roc_auc_score(y_true, y_pred_)

        self.f1_micro.append(score)
        print(" F1 micro :",score)

metrics=f1_score_and_auc_Callback()
```

In [ ]:

```
tf.config.experimental_run_functions_eagerly(True)
```

```
WARNING:tensorflow:From <ipython-input-21-bdb3352f611a>:1: experimental_run_functions_eagerly (from tensorflow.python.eager.def_functio
n) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.config.run_functions_eagerly` instead of the experimental version.
```

In [ ]:

```
earlyStopping = EarlyStopping(monitor='val_loss', patience=5, verbose=0, mode='min')
best_model = ModelCheckpoint('best_model_1.h5', save_best_only=True, monitor='val_loss', mode='min')
callback_list = [metrics,best_model,earlyStopping]
model.fit(X_train2,y_train,epochs=9, validation_data=(X_test2,y_test),callbacks=callback_list)
```

```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
Epoch 1/15
442/442 [==============================] - 339s 751ms/step - loss: 2.8804 - accuracy: 0.0802 - val_loss: 1.9814 - val_accuracy: 0.2902
 F1 micro : 0.07117059698321648
Epoch 2/15
442/442 [==============================] - 332s 752ms/step - loss: 2.2119 - accuracy: 0.2645 - val_loss: 1.8224 - val_accuracy: 0.3433
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.13001912045889102
Epoch 3/15
442/442 [==============================] - 331s 750ms/step - loss: 1.5884 - accuracy: 0.4270 - val_loss: 1.4448 - val_accuracy: 0.4793
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.26704907584448695
Epoch 4/15
442/442 [==============================] - 331s 748ms/step - loss: 0.9996 - accuracy: 0.6137 - val_loss: 1.2884 - val_accuracy: 0.5626
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.41470150839175696
Epoch 5/15
442/442 [==============================] - 330s 746ms/step - loss: 0.6886 - accuracy: 0.7380 - val_loss: 1.3111 - val_accuracy: 0.6223
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.5389844911833439
Epoch 6/15
442/442 [==============================] - 329s 745ms/step - loss: 0.5101 - accuracy: 0.8307 - val_loss: 1.3221 - val_accuracy: 0.6847
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.6347992351816444
Epoch 7/15
442/442 [==============================] - 329s 744ms/step - loss: 0.3132 - accuracy: 0.8959 - val_loss: 1.4336 - val_accuracy: 0.6913
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.6564690885914596
Epoch 8/15
442/442 [==============================] - 330s 746ms/step - loss: 0.2459 - accuracy: 0.9211 - val_loss: 1.4852 - val_accuracy: 0.7240
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.7051200339919269
Epoch 9/15
442/442 [==============================] - 329s 744ms/step - loss: 0.1612 - accuracy: 0.9456 - val_loss: 1.4244 - val_accuracy: 0.7140
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.6940726577437859
```

Out[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7fc3ad934048>
```

Model-2 : Using 1D convolutions with character embedding

In [ ]:

```
from tensorflow import keras
from keras.preprocessing.text import Tokenizer
```

```
import numpy as np

X_train_data = []
for i in i_train:
  string = ''
  for word in X_train[i].split():
    string = string + word
  X_train_data.append(string)

X_test_data = []
for i in i_test:
  string = ''
  for word in X_test[i].split():
    string = string + word
  X_test_data.append(string)

t = Tokenizer(filters='!"#$%&()*+,-./:;<=>?@[\\]^`{|}~\t\n', char_level=True, oov_token=True)
t.fit_on_texts(X_train_data)
X_train_token = np.array(t.texts_to_sequences(X_train_data))
X_test_token = np.array(t.texts_to_sequences(X_test_data))
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:23: VisibleDeprecationWarning: Creating an ndarray from ragged nested seque
nces (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this,
you must specify 'dtype=object' when creating the ndarray
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:24: VisibleDeprecationWarning: Creating an ndarray from ragged nested seque
nces (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this,
you must specify 'dtype=object' when creating the ndarray
```

In [ ]:

```
li = []
for concat in X_train_data:
  li.append(len(concat))
print(max(li))
```

```
43825
```

In [ ]:

```
voc_size = len(t.word_index) + 1
voc_size
```

Out[ ]:

```
29
```

In [ ]:

```
X_train2 = sequence.pad_sequences(X_train_token, maxlen=43825)
X_test2 = sequence.pad_sequences(X_test_token, maxlen=43825)
```

In [ ]:

```
from tensorflow.keras.layers import Dense,Input,Conv1D,MaxPooling1D,Activation,Dropout,Flatten
from tensorflow.keras.models import Model
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers.embeddings import Embedding
from keras.preprocessing import sequence
import random as rn
import keras
import os
os.environ['PYTHONHASHSEED'] = '0'

##https://keras.io/getting-started/faq/#how-can-i-obtain-reproducible-results-using-keras-during-development
## Have to clear the session. If you are not clearing, Graph will create again and again and graph size will increses.
## Varibles will also set to some value from before session
tf.keras.backend.clear_session()

## Set the random seed values to regenerate the model.
np.random.seed(0)
rn.seed(0)

embedding_vecor_length = 30


inputs = Input(shape=(max(li),), dtype='int32', name='inputs')
embedding = Embedding(voc_size, embedding_vecor_length, input_length=max(li))(inputs)

con1 = Conv1D(filters=16, kernel_size=16, activation='relu')(embedding)
con2 = Conv1D(filters=16, kernel_size=16, activation='relu')(embedding)
con3 = Conv1D(filters=16, kernel_size=16, activation='relu')(embedding)

combine_con1 = keras.layers.concatenate([con1, con2,con3])

max_pool1 = MaxPooling1D(pool_size=2)(combine_con1)

con4 = Conv1D(filters=16, kernel_size=16, activation='relu')(max_pool1)
con5 = Conv1D(filters=16, kernel_size=16, activation='relu')(max_pool1)
con6 = Conv1D(filters=16, kernel_size=16, activation='relu')(max_pool1)

combine_con2 = keras.layers.concatenate([con4, con5,con6])


max_pool2 = MaxPooling1D(pool_size=2)(combine_con2)

con7 = Conv1D(filters=16, kernel_size=16, activation='relu')(max_pool2)

flatten = Flatten()(con7)
```

```
drop_out = Dropout(0.4)(flatten)

dense1 = Dense(40, activation='relu')(drop_out)

Out = Dense(units=20,activation='softmax')(dense1)

model = Model(inputs=inputs,outputs=Out)
```

In [ ]:

```
optimizer = tf.keras.optimizers.Adam(
    learning_rate=0.001,
    beta_1=0.9,
    beta_2=0.999,
    epsilon=1e-07,
    amsgrad=False,
    name="Adam",

)

model.compile(optimizer=optimizer,loss='categorical_crossentropy',metrics=['accuracy'])

earlyStopping = EarlyStopping(monitor='val_loss', patience=5, verbose=0, mode='min')
best_model = ModelCheckpoint('best_model_1.h5', save_best_only=True, monitor='val_loss', mode='min')
callback_list = [metrics,best_model,earlyStopping]
model.fit(X_train2,y_train,epochs=9, validation_data=(X_test2,y_test),callbacks=callback_list)
```

```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
Epoch 1/9
442/442 [==============================] - 458s 1s/step - loss: 2.9851 - accuracy: 0.0585 - val_loss: 2.9257 - val_accuracy: 0.0871
 F1 micro : 0.0
Epoch 2/9
442/442 [==============================] - 458s 1s/step - loss: 2.8966 - accuracy: 0.0967 - val_loss: 2.6669 - val_accuracy: 0.1181
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.0031867431485022306
Epoch 3/9
442/442 [==============================] - 454s 1s/step - loss: 2.6521 - accuracy: 0.1356 - val_loss: 2.6689 - val_accuracy: 0.1419
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.014021669853409816
Epoch 4/9
442/442 [==============================] - 456s 1s/step - loss: 2.4459 - accuracy: 0.1833 - val_loss: 2.3925 - val_accuracy: 0.1863
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.020607760569364776
Epoch 5/9
442/442 [==============================] - 456s 1s/step - loss: 2.2654 - accuracy: 0.2255 - val_loss: 2.3057 - val_accuracy: 0.2182
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.04227745910346293
Epoch 6/9
442/442 [==============================] - 457s 1s/step - loss: 2.0936 - accuracy: 0.2698 - val_loss: 2.2645 - val_accuracy: 0.2411
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.06862120246441471
Epoch 7/9
442/442 [==============================] - 455s 1s/step - loss: 1.9010 - accuracy: 0.3362 - val_loss: 2.1963 - val_accuracy: 0.2779
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.08816656044189505
Epoch 8/9
442/442 [==============================] - 452s 1s/step - loss: 1.7061 - accuracy: 0.3981 - val_loss: 2.1111 - val_accuracy: 0.2979
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.12385808370512004
Epoch 9/9
442/442 [==============================] - 451s 1s/step - loss: 1.5096 - accuracy: 0.4623 - val_loss: 2.1714 - val_accuracy: 0.3185
/usr/local/lib/python3.6/dist-packages/tensorflow/python/data/ops/dataset_ops.py:3504: UserWarning: Even though the tf.config.experiment
al_run_functions_eagerly option is set, this option does not apply to tf.data functions. tf.data functions are still traced and executed
as graphs.
  "Even though the tf.config.experimental_run_functions_eagerly "
 F1 micro : 0.1701720841300191
```

Out[ ]:

```
<tensorflow.python.keras.callbacks.History at 0x7f47302055c0>
```