Text Classification Assignment

May 5, 2021

1 Text Classification:

1.1 Data

sample document

1.1.1 Preprocessing:

[ ]:
```
# we have collected all emails and preprocessed them, this is sample output
preprocessed_email
```

[ ]:
```
array(['juliet caltech edu',
       'coding bchs edu newsgate sps mot austlcm sps mot austlcm sps mot com
dna bchs edu',
       'batman bmd trw', …, 'rbdc wsnc org dscomsa desy zeus  desy',
       'rbdc wsnc org morrow stanford edu pangea Stanford EDU',
       'rbdc wsnc org apollo apollo'], dtype=object)
```

[ ]:
```
len(preprocessed_email)
```

[ ]:
```
18828
```

[ ]:
```
data.columns
```

```
Index(['text', 'class', 'preprocessed_text', 'preprocessed_subject',
       'preprocessed_emails'],
      dtype='object')
```

[ ]:
```
data.iloc[400]
```

```
text                   From: arc1@ukc.ac.uk (Tony Curtis)\r\r\r\nSubj…
class                                                        alt.atheism
preprocessed_text      said re is article if followed the quoting rig…
preprocessed_subject                                christian morality is
preprocessed_emails                                   ukc mac macalstr edu
Name: 567, dtype: object
```

[ ]:
```
import nltk
nltk.download("punkt")
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Dell\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Dell\AppData\Roaming\nltk_data…
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     C:\Users\Dell\AppData\Roaming\nltk_data…
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
[nltk_data] Downloading package words to
[nltk_data]     C:\Users\Dell\AppData\Roaming\nltk_data…
[nltk_data]   Package words is already up-to-date!
```

[ ]:
```
True
```

[ ]:
```
import regex as re
```

[ ]:
```
import os
import regex as re
from bs4 import BeautifulSoup
from nltk import ne_chunk, pos_tag, word_tokenize
from nltk.tree import Tree
```

[ ]:
```
def preprocess(file):
    """Do all the Preprocessing as shown above and
    return a tuple contain preprocess_email,preprocess_subject,preprocess_text␣
 ↪for that Text_data"""


    class_ = file.split('_')[0]

    with open("documents/"+file,'rb') as f:
        text = f.read()
        orignal_data = text


        ## Remove tags
        soup = BeautifulSoup(text,'lxml')
        text = soup.get_text()
```

```
## email
emails = re.findall("[\w-]+@[\w\.-]+",text)

process_emails = []
for e in emails:
    email = e.split("@")[1].split(".")
    pro  = [s for s in email if len(s)>2  and s.lower()!='com']
    process_emails.extend(pro)

final_mail = ""
for i in process_emails:
    final_mail += " "+i

## Subject
if(re.findall(r'Subject:.*',text)):
    subject = re.findall(r'Subject:.*',text)
    subject = subject[0].split(":")[-1]
    subject = re.sub('[^A-Za-z0-9]+',' ',subject)
    subject = subject.lower()
else:
    subject = " "


## removing subjects and emails
text = re.sub(r'Subjects:.*',"",text)
text = re.sub("[\w-]+@[\w\.-]+","",text)


## Deleting all the sentances where sentence starts
## with "Write to:" or "From:".

text = re.sub(r'Write to:.*',"",text)
text = re.sub(r'From:.*',"",text)


## removing new line, tabs, '-',"\"
text = re.sub(r"\s+"," ",text)
text = re.sub(r"[/]",".",text)

## Remove words ending with ":"
text = re.sub(r"[a-zA-Z]+:"," ",text)


## Decontractions
text = re.sub(r"can\'t","can not",text)
text = re.sub(r"\'s","is",text)
```

```
        text = re.sub(r"i\'ve'","i have",text)
        text = re.sub(r"i\'m","i am",text)
        text = re.sub(r"you\'re","you are",text)
        text = re.sub(r"i\'ll","i will",text)
        text = re.sub(r"\'d","would",text)


        ## Chunking
        chunks = ne_chunk(pos_tag(word_tokenize(text)),binary=True)

        for i in chunks:
            if type(i)==Tree:
                if i.label()=='PERSON':
                    for j,k in i.leaves():
                        text = re.sub(r'\b{}\b'.format(j)," ",text,count=1)

                else:
                    string = ""
                    chunked_string = ""

                    for j,k in i.leaves():
                        string += j + " "
                        chunked_string += j + "_"


                    string = string.strip()
                    chunked_string = chunked_string[:-1]

                    text = re.sub(r"\b{}\b".
↪format(string),chunked_string,text,count=1)

        text = re.sub(r"\s+"," ",text)



        ## Delete Number

        text = re.sub(r"[0-9]","",text)

        ## Delete  _word_ type words

        text = re.sub(r"(_?)([A-Za-z0-9])(_?)",r'\2',text)

        ## delete oneletter word  and two letter word

        text = re.sub(r"([A-Za-z]{1,2})(_)(A-Za-z)","\g<3>",text)
```

```
        ## Replace all word except A-Za-z_

        text = re.sub(r'[^A-Za-z_]'," ",text)

        ## Lowe case and remove 2 >  len(text), len(text) > 15

        text =text.lower()
        text = ' '.join([w for w in text.split() if len(w)>2 and len(w)<15])



    return [orignal_data,class_,final_mail,subject,text]
```

[ ]:
```
import time
start  = time.time()
lst = preprocess('alt.atheism_49960.txt')
end = time.time()
print(end-start)
```

```
1.1414048671722412
```

[ ]:
```
lst
```

[ ]:
```
[b'From: mathew <mathew@mantis.co.uk>\nSubject: Alt.Atheism FAQ: Atheist
Resources\n\nArchive-name: atheism/resources\nAlt-atheism-archive-name:
resources\nLast-modified: 11 December 1992\nVersion: 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\nEVOLUTION 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:  Evolution Designs, 7119 Laurel Canyon
#4, North Hollywood,\n          CA 91605.\n\nPeople in the San Francisco Bay
area can get Darwin Fish from Lynn Gold --\ntry mailing <figmo@netcom.com>.  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 is:\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.  Bible contradictions,\nabsurdities,
atrocities, immoralities… contains Ball, Foote: "The Bible\nContradicts
Itself", AAP.  Based on the King James version of the Bible.\n\nWrite to:
```

American Atheist Press, P.O. Box 140195, Austin, TX 78714-0195.\n or: 7215 Cameron Road, Austin, TX 78752-2973.\nTelephone: (512) 458-1244\nFax: (512) 467-9525\n\nPROMETHEUS BOOKS\n\nSell books including Haught\'s "Holy Horrors" (see below).\n\nWrite to: 700 East Amherst Street, Buffalo, New York 14215.\nTelephone: (716) 837-2475.\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 publish 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 Society\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 1271 071 831 7723\n\nThe National Secular Society publish "The Freethinker", a monthly magazine\nfounded in 1881.\n\n Germany\n\nIBKA e.V.\nInternationaler Bund der Konfessionslosen und Atheisten\nPostfach 880, D-1000 Berlin 41. Germany.\n\nIBKA publish a journal:\nMIZ. (Materialien und Informationen zur Zeit. Politisches\nJournal der Konfessionslosesn und Atheisten. Hrsg. IBKA e.V.)\nMIZ-Vertrieb, Postfach 880, D-1000 Berlin 41. Germany.\n\nFor atheist books, write to:\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 ultimate proof that Santa exists. All characters and \nevents are fictitious. Any similarity to living or dead gods -- uh, well…\n\nWALTER M. MILLER, JR\n\n"A Canticle for Leibowitz"\nOne gem in this post atomic doomsday novel is the monks who spent their lives\ncopying 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 short \nstories and novels. His stories are bizarre at times, but very approachable.\nHe wrote mainly SF, but he wrote about people, truth and religion rather than\ntechnology. Although he often believed that he had met some sort of God, he\nremained sceptical. Amongst his novels, the following are of some relevance:\n\n"Galactic Pot-Healer"\nA fallible alien deity summons a group of Earth craftsmen 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 for its description of a technology-based religion.\n\n"VALIS"\nThe schizophrenic hero searches for the hidden mysteries of Gnostic\nChristianity after reality is fired into his brain by a pink laser beam of\nunknown but possibly divine origin. He is accompanied by his 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 premise that the US Congress is mysteriously\nassassinated, and fundamentalists quickly take charge of the nation to set it\n"right" again. 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 property is revoked,\nand their bank accounts are closed; sinful luxuries are outlawed, and the\nradio is only used for readings from the Bible. Crimes are punished\nretroactively: doctors who performed legal abortions in the "old world" are\nhunted down and hanged. Atwood\'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 AUTHORS\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 version.\n\n Books -- Non-fiction\n\nPETER DE ROSA\n\n"Vicars of Christ", Bantam Press, 1988\nAlthough de Rosa seems to be Christian or even Catholic this is a very\nenlighting history of papal immoralities, adulteries, fallacies etc.\n(German translation: "Gottes erste Diener. Die dunkle Seite des Papsttums",\nDroemer-Knaur, 1989)\n\nMICHAEL MARTIN\n\n"Atheism: A Philosophical Justification", Temple University Press,\n Philadelphia, USA.\nA detailed and scholarly justification of atheism. Contains an 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)"). Includes great refutations of the most\nchallenging arguments for god; particular attention is paid to refuting\ncontempory theists such 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 Christianity 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 America". Examines the way in which\nunbelief (whether agnostic or atheistic) became a mainstream alternative\nworld-view. Focusses on the period 1770-1900, and while considering France\nand Britain the emphasis is on American, and particularly New England\ndevelopments. "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 SELDES (Editor)\n\n"The great thoughts", Ballantine Books, New York, USA\nA "dictionary of quotations" of a different kind, concentrating 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 and twisted the idea over the centuries. Quite a number of\nthe quotations are derived from Cardiff\'s "What Great Men Think of Religion"\nand Noyes\' "Views of Religion".\n490 pages. ISBN (paper) 0-345-29887-X.\n\nRICHARD SWINBURNE\n\n"The Existence of God (Revised 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 arguments 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 somewhat incoherent attempt to rebut Mackie.\n\nJ. L. MACKIE\n\n"The Miracle of Theism", Oxford\nThis (posthumous) volume contains a comprehensive review of the principal\narguments for and against the existence of God. It ranges from the classical\nphilosophical positions 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 realm 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 with the hand-waving of Swinburne.\n\nJAMES A. HAUGHT\n\n"Holy Horrors: An Illustrated History of Religious Murder and Madness",\n Prometheus Books\nLooks at religious persecution from ancient times to the present day -- and\nnot only by Christians.\nLibrary of Congress Catalog Card Number 89-64079. 1990.\n\nNORM R. ALLEN, JR.\n\n"African American Humanism: an Anthology"\nSee the listing for African Americans for Humanism above.\n\nGORDON STEIN\n\n"An Anthology of Atheism and Rationalism", Prometheus Books\nAn anthology covering 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 fundamentalists, and what effect it\nhas on them.\n\n
Net Resources\n\nThere\'s a small mail-based archive server at mantis.co.uk which carries\narchives of old alt.atheism.moderated articles and assorted other files. For\nmore information, send mail to archive-server@mantis.co.uk saying\n\n   help\n   send atheism/index\n\nand it will mail back a reply.\n\n\nmathew\n\xff\n',
 'alt.atheism',
 ' mantis',
 ' atheist resources',
 'alt atheism atheist resources archive atheism resources alt atheism archive resources last december usa freedom from religion foundation darwin fish bumper stickers and assorted other atheist paraphernalia are available from the the

evolution designs evolution designs sell the darwin fish itis fish symbol like the ones christians stick their cars but with feet and the word darwin written inside the deluxe moulded plastic fish postpaid the people the area can get darwinfish from lynngold try mailing for net people who lynn directly the price per fish american atheist press aap publish various atheist books critiques the bible lists biblical contradictions and one such book the biblehandbook ball and foote isbn edition bible contradictions absurdities atrocities immoralities contains ball the aap based the king james version the bible cameronroad austin prometheus books sell books including haughtis holy horrors see below alternate address which may newer older glenn drive buffalo african americans for humanism organization promoting black secular humanism and uncovering the history black freethought they publish quarterly newsletter aahexaminer buffalo press association islingtonhigh street holloway road london londonn society lambis conduit passage conway hall london wcr red lion square london wcr fax the national secular society publish the freethinker monthly magazine founded germany ibka der und atheisten postfach berlin germany ibka publish miz materialien und informationen zur zeit politisches journal der und atheisten hrsg ibka miz vertrieb postfach berlin germany for atheist books write ibdk ucherdienst der postfach hannover germany books fiction thomas disch the short story the ultimate proof that santa exists all characters and events are fictitious any similarity living dead gods well walterm miller canticle for leibowitz one gem this post atomic doomsday novel the monks who spent their lives copying blueprints from saint leibowitz filling the sheets paper with ink and leaving white lines and letters edgar pangborn davy post atomic doomsday novel set clerical states the church for example forbids that anyone produce describe use any substance containing atoms philip dick philip dick dick wrote many philosophical and thought provoking short stories and novels his stories are bizarre times but very approachable wrote mainly but wrote about people truth and religion rather than technology although often believed that had met some sort god remained sceptical amongst his novels the following are some galactic pot healer fallible alien deity summons group earth craftsmen and women remote planet raise giant cathedral from beneath the oceans when the deity begins demand faith from the earthers pot healer joefernwright unable comply polished ironic and amusing novel maze death noteworthy for its description technology based religion valis the schizophrenic hero searches for the hidden mysteries after reality fired into his brain pink laser beam unknown but possibly divine origin accompanied his dogmatic and dismissively atheist friend and assorted other odd characters the divineinvasion god invades earth making young woman pregnant she returns from another star system unfortunately she terminally ill and must assisted dead man whose brain wired hour easy listening music margaret atwood the handmaidistale story based the premise that the uscongress mysteriously assassinated and quickly take charge the nation set right again the book the diary womanis life she tries live under the new christian theocracy womenis right own property revoked and their bank accounts are closed sinful luxuries are outlawed and the radio only used for readings from the bible crimes are punished doctors who performed legal abortions the old world are hunted down and hanged atwoodis writing style difficult get used first

but the tale grows more and more chilling goes various authors the bible this somewhat dull and rambling work has often been criticized however probably worth reading only that you know what all the fuss about exists many different versions make sure you get the one true version books non fiction peterde rosa vicars christ bantampress although rosa seems christian even catholic this very enlighting history papal immoralities adulteries fallacies etc german gottes erste diener die dunkle seite des papsttums droemer knaur michael martin philosophical justification philadelphia usa detailed and scholarly justification atheism contains outstanding appendix defining terminology and usage this necessarily tendentious area argues both for negative atheism the non belief the existence god and also for positive atheism the belief the non existence god includes great refutations the most challenging arguments for god particular attention paid refuting contempory theists such platinga and swinburne pages isbn hardcover paperback also available the press comprehensive critique christianity which considers the best contemporary defences christianity and ultimately demonstrates that they are unsupportable and incoherent pages isbn james turner without god withoutcreed the baltimore usa subtitled the origins unbelief america examines the way which unbelief whether agnostic atheistic became mainstream alternative world view focusses the period and while considering france and britain the emphasis american and particularly newengland developments neither religious history secularization atheism without god withoutcreed rather the intellectual history the fate single idea the belief that god exists pages isbn hardcover paper george seldes editor the great thoughts newyork usa dictionary quotations different kind concentrating statements and writings which explicitly implicitly present the personis philosophy and world view includes obscure and often suppressed opinions from many people for some popular observations traces the way which various people expressed and twisted the idea over the centuries quite number the quotations are derived from cardiffis what greatmenthink religion and noyes views religion pages isbn paper richard swinburne the existence god revised edition oxford this book the second volume trilogy that began with the coherence theism and was concluded with faith and reason this work swinburne attempts construct series inductive arguments for the existence god his arguments which are somewhat tendentious and rely upon the imputation late century western christian values and aesthetics god which supposedly simple can conceived were decisively rejected mackieis the miracle theism the revised edition the existence god swinburne includes appendix which makes somewhat incoherent attempt rebut mackie mackie the miracle theism oxford this posthumous volume contains comprehensive review the principal arguments for and against the existence god ranges from the classical philosophical positions descartes anselm berkeley hume through the moral arguments newman kant and sidgwick the recent restatements the classical theses plantinga and swinburne also addresses those positions which push the concept god beyond the realm the rational such those kierkegaard kung and philips well replacements for god such lelieis axiarchism the book delight read less formalistic and better written than martinis works and refreshingly direct when compared with the hand waving swinburne james haught holy illustrated history and madness religious persecution from ancient times the present day and

```
not only christians library number norm allen anthology see the listing for for
humanism above gordonstein anthology atheism and rationalism anthology covering
wide range subjects including the devil evil and morality and the history
freethought comprehensive bibliography edmund cohen the mind the bible believer
prometheus books study why people become christian and what effect has them net
resources thereis small mail based archive server mantis which carries archives
old alt atheism moderated articles and assorted other files for more information
send mail saying help send atheism index and will mail back reply mathew']
```

[ ]:
```
## Preprocess of  every file
```

[ ]:
```
row = []
done = 0
for f in os.listdir('documents'):
    if done%500==0:
        print(done)
    done+=1

    lst = preprocess(f)
    row.append(lst)
```

```
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
6000
6500
7000
7500
8000
8500
9000
9500
10000
10500
```

```
11000
11500
12000
12500
13000
13500
14000
14500
15000
15500
16000
16500
17000
17500
18000
18500
```

[ ]:
```
data = pd.DataFrame(row,columns =␣
↪["text","class","email","subject","preprocessed_text"])
```

[ ]:
```
data.head(2)
```

[ ]:
```
                                                text        class    email  \
0  b'From: mathew <mathew@mantis.co.uk>\nSubject:…  alt.atheism   mantis
1  b'From: mathew <mathew@mantis.co.uk>\nSubject:…  alt.atheism   mantis

                     subject                                    preprocessed_text
0         atheist resources  alt atheism atheist resources archive atheism …
1   introduction to atheism  alt atheism introduction atheism archive athei…
```

[ ]:
```
data.to_pickle("./preprocessed_data")
```

[ ]:
```
from google.colab import drive
```

[ ]:
```
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id
=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redire
ct_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&response_type=code&scope=email%20http
s%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.c
om%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.reado
nly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

Enter your authorization code:
··········
Mounted at /content/drive
```

[ ]:
```
cd /content/drive/My Drive
```

```
/content/drive/My Drive
```

[ ]:
```
cd 22) CNN ON TEXT DATA
```

```
/content/drive/My Drive/22) CNN ON TEXT DATA
```

[ ]:
```
import pandas as pd
```

[ ]:
```
my_dataframe = pd.read_pickle("preprocessed_data")
```

[ ]:
```
my_dataframe.head()
```

[ ]:
```
                                                text  …
preprocessed_text
0  b'From: mathew <mathew@mantis.co.uk>\nSubject:…  …  alt atheism atheist
resources archive atheism …
1  b'From: mathew <mathew@mantis.co.uk>\nSubject:…  …  alt atheism
introduction atheism archive athei…
2  b'From: I3150101@dbstu1.rz.tu-bs.de (Benedikt …  …  gospeldating article
charleywingate well john …
3  b'From: mathew <mathew@mantis.co.uk>\nSubject:…  …  university violating
separation church state u…
4  b'From: strom@Watson.Ibm.Com (Rob Strom)\nSubj…  …  soc motss princeton
axes matching funds for bo…

[5 rows x 5 columns]
```

[ ]:
```
my_dataframe.shape
```

[ ]:
```
(18828, 5)
```

[ ]:
```
cols = ['email', 'subject', 'preprocessed_text']
my_dataframe['total_data'] = my_dataframe[cols].apply(lambda row: ' '.join(row.
↪values.astype(str)), axis=1)
```

[ ]:
```
my_dataframe.head(2)
```

[ ]:
```
                                                text  …
total_data
0  b'From: mathew <mathew@mantis.co.uk>\nSubject:…  …   mantis  atheist
resources alt atheism atheist…
1  b'From: mathew <mathew@mantis.co.uk>\nSubject:…  …   mantis  introduction
to atheism alt atheism i…

[2 rows x 6 columns]
```

[ ]:
```
model_data = my_dataframe[['class','total_data']]
```

[ ]:
```
model_data.head(2)
```

[ ]:
```
          class                                          total_data
0  alt.atheism   mantis  atheist resources alt atheism atheist…
1  alt.atheism   mantis  introduction to atheism alt atheism i…
```

1.1.2 Training The models to Classify:

1.1.3 Model-1: Using 1D convolutions with word embeddings

https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/

ref: 'https://i.imgur.com/fv1GvFJ.png'

1.1.4 Model-2 : Using 1D convolutions with character embedding

[ ]:
```
%load_ext tensorboard
```

[ ]:
```
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Model
from tensorflow.keras.layers import␣
↪Input,Dense,Conv1D,Flatten,Embedding,MaxPool1D,concatenate,Dropout
from tensorflow.keras.callbacks import ModelCheckpoint,TensorBoard,EarlyStopping
from tensorflow.keras.optimizers import Adam
```

[ ]:
```
x = model_data['total_data']
y = model_data['class']
from sklearn.preprocessing import LabelEncoder
from keras.utils import np_utils
## encoding lables
encoder = LabelEncoder()
encoder.fit(y)
encoder_y = encoder.transform(y)
## converting it to a matrix
y = np_utils.to_categorical(encoder_y)
```

[ ]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,stratify=y)
```

[ ]:
```
## Finding the sequence length
```

[31]:
```
length_sentence =[len(s) for s in x_train]
length_sentence.sort()
```

```
length_sentence = np.array(length_sentence)
```

[ ]:
```
percentile_95 = int(np.percentile(length_sentence,95))
percentile_95
```

[ ]:
```
3452
```

[ ]:
```
percentile_98 = int(np.percentile(length_sentence,98))
percentile_98
```

[ ]:
```
5960
```

[ ]:
```
maxl = percentile_98
```

Since 98% of the size of the sentences are less than 6231 we will use maxlen = 6231

[ ]:
```
tokenizer = Tokenizer(filters='!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~\t\n')
tokenizer.fit_on_texts(x_train)
x_train  = tokenizer.texts_to_sequences(x_train)
x_test = tokenizer.texts_to_sequences(x_test)
```

[ ]:
```
x_train = pad_sequences(x_train,maxlen=maxl,padding="post")
x_test= pad_sequences(x_test,maxlen=maxl,padding='post')
```

[ ]:
```
print(x_train.shape)
print(x_test.shape)
```

```
(14121, 5960)
(4707, 5960)
```

[ ]:
```
##https://medium.com/analytics-vidhya/
 ↪basics-of-using-pre-trained-glove-vectors-in-python-d38905f356db

embedding_dict = {}
pretrain =  open("glove.6B.50d.txt")
for i in pretrain:
    value = i.split(" ")
    word = value[0]
    vector = np.asarray(value[1:])
    embedding_dict[word] = vector
pretrain.close()
```

[ ]:
```
len(tokenizer.index_word)
```

[ ]: 99006

[ ]:
```
## Converting embedding word to embedding matrix

import numpy as np

size = len(tokenizer.word_index)+1

emb_matrix = np.zeros((size,50))

for word,i in tokenizer.word_index.items():
    emb_word = embedding_dict.get(word)

    if emb_word is not None:
      emb_matrix[i]=emb_word
```

[ ]:
```
print(emb_matrix.shape)
```

```
(99007, 50)
```

[ ]:
```
size
```

[ ]: 99007

Model1

[ ]:
```
pip install -U tensorflow-estimator
```

```
Requirement already up-to-date: tensorflow-estimator in
/usr/local/lib/python3.6/dist-packages (2.2.0)
```

[ ]:
```
## Embedding layer
embedding_layer = Embedding(len(tokenizer.word_index)+1, 50,␣
 ↪embeddings_initializer=tf.keras.initializers.
 ↪Constant(emb_matrix),trainable=False)
```

[ ]:
```
## Defining layer
```

[ ]:
```
first_layer = Input(shape=(maxl))

embed = embedding_layer(first_layer)

m1 = Conv1D(32,4,activation="relu",kernel_initializer =tf.keras.initializers.
 ↪he_normal(),kernel_regularizer=tf.keras.regularizers.l2())(embed)
```

```
n1 = Conv1D(32,4,activation="relu",kernel_initializer =tf.keras.initializers.
 ↪he_normal(),kernel_regularizer=tf.keras.regularizers.l2())(embed)

o1 = Conv1D(32,4,activation="relu",kernel_initializer = tf.keras.initializers.
 ↪he_normal(),kernel_regularizer=tf.keras.regularizers.l2())(embed)

second_layer = concatenate([m1,n1,o1])

max_pool_1 = MaxPool1D(3)(second_layer)

i1 = Conv1D(32,3,activation="relu",kernel_initializer = tf.keras.initializers.
 ↪he_normal(),kernel_regularizer=tf.keras.regularizers.l2())(max_pool_1)

j1 = Conv1D(32,3,activation="relu",kernel_initializer = tf.keras.initializers.
 ↪he_normal(),kernel_regularizer=tf.keras.regularizers.l2())(max_pool_1)

k1 =  Conv1D(32,3,activation="relu",kernel_initializer = tf.keras.initializers.
 ↪he_normal(),kernel_regularizer=tf.keras.regularizers.l2())(max_pool_1)

third_layer = concatenate([i1,j1,k1])

max_pool_2 = MaxPool1D(3)(third_layer)

fourth_layer = Conv1D(32,3,activation='relu',
                      kernel_initializer = tf.keras.initializers.
 ↪he_normal(seed=42),kernel_regularizer=tf.keras.regularizers.l2())(max_pool_2)

flatten = Flatten()(fourth_layer)

dropout_layer = Dropout(0.2)(flatten)

dense_layer = Dense(64,activation="relu",kernel_initializer = tf.keras.
 ↪initializers.he_normal())(dropout_layer)

output_layer = Dense(20,activation="softmax",kernel_initializer= tf.keras.
 ↪initializers.glorot_normal())(dense_layer)


model =Model(inputs=first_layer,outputs=output_layer)
```

[ ]:
```
model.summary()
```

```
Model: "model_6"
_________________________________________________________________________________
__________________
```

```
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================
==================
input_6 (InputLayer)            [(None, 5960)]       0
----------------------------------------------------------------------------------
------------------
embedding_3 (Embedding)         (None, 5960, 50)     4950350     input_6[0][0]
----------------------------------------------------------------------------------
------------------
conv1d_38 (Conv1D)              (None, 5957, 32)     6432
embedding_3[2][0]
----------------------------------------------------------------------------------
------------------
conv1d_39 (Conv1D)              (None, 5957, 32)     6432
embedding_3[2][0]
----------------------------------------------------------------------------------
------------------
conv1d_40 (Conv1D)              (None, 5957, 32)     6432
embedding_3[2][0]
----------------------------------------------------------------------------------
------------------
concatenate_4 (Concatenate)     (None, 5957, 96)     0           conv1d_38[0][0]
                                                                 conv1d_39[0][0]
                                                                 conv1d_40[0][0]
----------------------------------------------------------------------------------
------------------
max_pooling1d_19 (MaxPooling1D) (None, 1985, 96)     0
concatenate_4[0][0]
----------------------------------------------------------------------------------
------------------
conv1d_41 (Conv1D)              (None, 1983, 32)     9248
max_pooling1d_19[0][0]
----------------------------------------------------------------------------------
------------------
conv1d_42 (Conv1D)              (None, 1983, 32)     9248
max_pooling1d_19[0][0]
----------------------------------------------------------------------------------
------------------
conv1d_43 (Conv1D)              (None, 1983, 32)     9248
max_pooling1d_19[0][0]
----------------------------------------------------------------------------------
------------------
concatenate_5 (Concatenate)     (None, 1983, 96)     0           conv1d_41[0][0]
                                                                 conv1d_42[0][0]
                                                                 conv1d_43[0][0]
----------------------------------------------------------------------------------
------------------
max_pooling1d_20 (MaxPooling1D) (None, 661, 96)      0
```

```
concatenate_5[0][0]
__________________________________________________________________________________
__________________
conv1d_44 (Conv1D)              (None, 659, 32)      9248
max_pooling1d_20[0][0]
__________________________________________________________________________________
__________________
flatten_6 (Flatten)             (None, 21088)        0           conv1d_44[0][0]
__________________________________________________________________________________
__________________
dropout_6 (Dropout)             (None, 21088)        0           flatten_6[0][0]
__________________________________________________________________________________
__________________
dense_12 (Dense)                (None, 64)           1349696     dropout_6[0][0]
__________________________________________________________________________________
__________________
dense_13 (Dense)                (None, 20)           1300        dense_12[0][0]
==================================================================================
==================
Total params: 6,357,634
Trainable params: 1,407,284
Non-trainable params: 4,950,350
__________________________________________________________________________________
__________________
```

[ ]:
```
import tensorflow_addons as tfa
from tensorflow_addons.metrics import F1Score
```

[ ]:
```
## f1_score_callback
custom_callback = custom()

## Callback for saving best model
checkpoint = ModelCheckpoint(filepath='best_model_1.
 ↪h5',verbose=1,monitor='val_accuracy',
                             mode='max',save_best_only=True)

## Callback for earlystopping
early_stop = EarlyStopping(monitor="val_accuracy",mode='max',patience=2)

## Tensorboard
log_dir = "logs"
tensorboard = TensorBoard(log_dir=log_dir,histogram_freq=1,write_graph=True)

## all callbacks
callbacks =[checkpoint,early_stop,tensorboard]

## compile model
```

```
model.compile(loss='categorical_crossentropy', optimizer=Adam(learning_rate=0.
 ↪001), metrics=['accuracy',F1Score(average='micro',num_classes=20)])


## Trainning


model.
 ↪fit(x_train,y_train,epochs=15,verbose=2,validation_data=(x_test,y_test),batch_size␣
 ↪=64,callbacks=callbacks)
```

```
Epoch 1/15

Epoch 00001: val_accuracy improved from -inf to 0.30763, saving model to
best_model_1.h5
221/221 - 48s - loss: 3.9078 - accuracy: 0.1861 - f1_score: 0.1861 - val_loss:
2.3858 - val_accuracy: 0.3076 - val_f1_score: 0.3076
Epoch 2/15

Epoch 00002: val_accuracy improved from 0.30763 to 0.40705, saving model to
best_model_1.h5
221/221 - 47s - loss: 2.1423 - accuracy: 0.3539 - f1_score: 0.3539 - val_loss:
1.9610 - val_accuracy: 0.4071 - val_f1_score: 0.4071
Epoch 3/15

Epoch 00003: val_accuracy improved from 0.40705 to 0.48375, saving model to
best_model_1.h5
221/221 - 47s - loss: 1.8327 - accuracy: 0.4410 - f1_score: 0.4410 - val_loss:
1.7151 - val_accuracy: 0.4837 - val_f1_score: 0.4837
Epoch 4/15

Epoch 00004: val_accuracy improved from 0.48375 to 0.54578, saving model to
best_model_1.h5
221/221 - 47s - loss: 1.6455 - accuracy: 0.5040 - f1_score: 0.5040 - val_loss:
1.5801 - val_accuracy: 0.5458 - val_f1_score: 0.5458
Epoch 5/15

Epoch 00005: val_accuracy improved from 0.54578 to 0.55980, saving model to
best_model_1.h5
221/221 - 47s - loss: 1.5229 - accuracy: 0.5496 - f1_score: 0.5496 - val_loss:
1.4866 - val_accuracy: 0.5598 - val_f1_score: 0.5598
Epoch 6/15

Epoch 00006: val_accuracy improved from 0.55980 to 0.59422, saving model to
best_model_1.h5
221/221 - 47s - loss: 1.4189 - accuracy: 0.5876 - f1_score: 0.5876 - val_loss:
1.4439 - val_accuracy: 0.5942 - val_f1_score: 0.5942
```

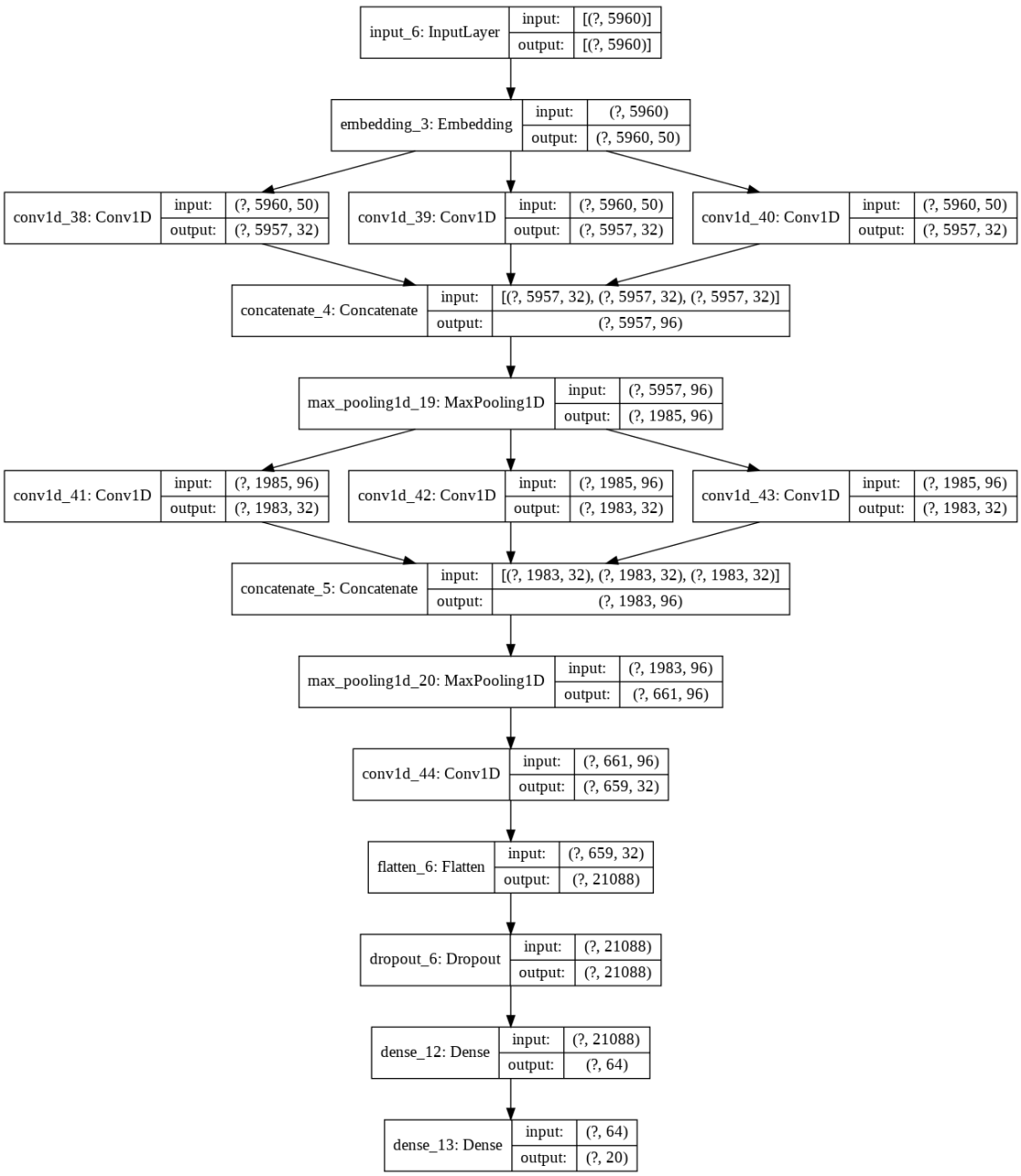
```
Epoch 7/15

Epoch 00007: val_accuracy improved from 0.59422 to 0.60229, saving model to
best_model_1.h5
221/221 - 46s - loss: 1.3649 - accuracy: 0.6073 - f1_score: 0.6073 - val_loss:
1.4130 - val_accuracy: 0.6023 - val_f1_score: 0.6023
Epoch 8/15

Epoch 00008: val_accuracy improved from 0.60229 to 0.63671, saving model to
best_model_1.h5
221/221 - 47s - loss: 1.2930 - accuracy: 0.6372 - f1_score: 0.6372 - val_loss:
1.3170 - val_accuracy: 0.6367 - val_f1_score: 0.6367
Epoch 9/15

Epoch 00009: val_accuracy did not improve from 0.63671
221/221 - 43s - loss: 1.2349 - accuracy: 0.6599 - f1_score: 0.6599 - val_loss:
1.4001 - val_accuracy: 0.6238 - val_f1_score: 0.6238
Epoch 10/15

Epoch 00010: val_accuracy improved from 0.63671 to 0.64585, saving model to
best_model_1.h5
221/221 - 47s - loss: 1.2086 - accuracy: 0.6720 - f1_score: 0.6720 - val_loss:
1.3032 - val_accuracy: 0.6458 - val_f1_score: 0.6458
Epoch 11/15

Epoch 00011: val_accuracy improved from 0.64585 to 0.67007, saving model to
best_model_1.h5
221/221 - 46s - loss: 1.1637 - accuracy: 0.6883 - f1_score: 0.6883 - val_loss:
1.2455 - val_accuracy: 0.6701 - val_f1_score: 0.6701
Epoch 12/15

Epoch 00012: val_accuracy did not improve from 0.67007
221/221 - 43s - loss: 1.1512 - accuracy: 0.6953 - f1_score: 0.6953 - val_loss:
1.2611 - val_accuracy: 0.6654 - val_f1_score: 0.6654
Epoch 13/15

Epoch 00013: val_accuracy improved from 0.67007 to 0.68855, saving model to
best_model_1.h5
221/221 - 47s - loss: 1.1112 - accuracy: 0.7140 - f1_score: 0.7140 - val_loss:
1.2066 - val_accuracy: 0.6885 - val_f1_score: 0.6885
Epoch 14/15

Epoch 00014: val_accuracy did not improve from 0.68855
221/221 - 43s - loss: 1.0872 - accuracy: 0.7252 - f1_score: 0.7252 - val_loss:
1.2872 - val_accuracy: 0.6696 - val_f1_score: 0.6696
Epoch 15/15

Epoch 00015: val_accuracy improved from 0.68855 to 0.69811, saving model to
```

```
best_model_1.h5
221/221 - 46s - loss: 1.0602 - accuracy: 0.7351 - f1_score: 0.7351 - val_loss:
1.2049 - val_accuracy: 0.6981 - val_f1_score: 0.6981
```

[ ]:
```
<tensorflow.python.keras.callbacks.History at 0x7f6330bcf198>
```

[ ]:
```
tf.keras.utils.plot_model(model,to_file = 'model1.
↪png',show_shapes=True,show_layer_names=True)
```

[ ]:



[ ]:
```
%tensorboard --logdir logs
```

```
Output hidden; open in https://colab.research.google.com to view.
```

Model 2

[ ]:
```
##https://towardsdatascience.com/
 ↪how-to-preprocess-character-level-text-with-keras-349065121089
```

[25]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25,stratify=y)
```

[26]:
```
tokenize_char = Tokenizer(filters='!"#$%&()*+,-./:;<=>?
 ↪@[\\]^_`{|}~\t\n',char_level= True,oov_token='UNK')
## train it on train
tokenize_char.fit_on_texts(x_train)
```

[27]:
```
print(tokenize_char.word_index)
```

```
{'UNK': 1, ' ': 2, 'e': 3, 't': 4, 'a': 5, 'o': 6, 'i': 7, 'n': 8, 's': 9, 'r':
10, 'h': 11, 'l': 12, 'd': 13, 'c': 14, 'u': 15, 'm': 16, 'p': 17, 'g': 18, 'y':
19, 'w': 20, 'f': 21, 'b': 22, 'v': 23, 'k': 24, 'x': 25, 'j': 26, 'z': 27, 'q':
28, '_': 29, '1': 30, '0': 31, '2': 32, '-': 33, '3': 34, '4': 35, '6': 36, '5':
37, '8': 38, '9': 39, '7': 40}
```

[28]:
```
size_char = len(tokenize_char.word_index)+1
print(size_char)
```

```
41
```

[29]:
```
## Tokenize them
x_train = tokenize_char.texts_to_sequences(x_train)
x_test = tokenize_char.texts_to_sequences(x_test)
```

[32]:
```
maxl = int(np.percentile(length_sentence,99))
```

[33]:
```
print(maxl)
```

```
9121
```

[34]:
```
x_train = pad_sequences(x_train,maxlen=maxl,padding="post")
x_test = pad_sequences(x_test,maxlen=maxl,padding="post")
print(f"x_train_shape{x_train.shape}")
print(f"x_train_shape{x_test.shape}")
```

```
x_train_shape(14121, 9121)
x_train_shape(4707, 9121)
```

[35]:
```
## Make a embedding matrix

emb_matrix_char = np.zeros((41,41))

#print(tokenize_char.word_index)
for i,j in tokenize_char.word_index.items():
  emb_matrix_char[j][j]=1
```

[36]:
```
print(emb_matrix_char)
```

```
[[0. 0. 0. … 0. 0. 0.]
 [0. 1. 0. … 0. 0. 0.]
 [0. 0. 1. … 0. 0. 0.]
 …
 [0. 0. 0. … 1. 0. 0.]
 [0. 0. 0. … 0. 1. 0.]
 [0. 0. 0. … 0. 0. 1.]]
```

[37]:
```
embedding_layer_char = Embedding(len(tokenize_char.word_index)+1,41,␣
 ↪embeddings_initializer=tf.keras.initializers.
 ↪Constant(emb_matrix_char),input_length=maxl,trainable=False)
```

[38]:
```
first_layer = Input(shape=(maxl))

embed = embedding_layer_char(first_layer)
```

[39]:
```
m1 = Conv1D(64,3,activation="relu",kernel_initializer =tf.keras.initializers.
 ↪he_normal(seed=42),kernel_regularizer=tf.keras.regularizers.l1())(embed)

n1 = Conv1D(64,3,activation="relu",kernel_initializer =tf.keras.initializers.
 ↪he_normal(seed=42),kernel_regularizer=tf.keras.regularizers.l1())(m1)


max_pool_1 = MaxPool1D(5)(n1)

o1 = Conv1D(64,3,activation="relu",kernel_initializer = tf.keras.initializers.
 ↪he_normal(seed=42),kernel_regularizer=tf.keras.regularizers.l1())(max_pool_1)

i1 = Conv1D(64,3,activation="relu",kernel_initializer = tf.keras.initializers.
 ↪he_normal(seed=42),kernel_regularizer=tf.keras.regularizers.l1())(o1)


max_pool_2 = MaxPool1D(5)(i1)

i1 = Conv1D(64,3,activation="relu",kernel_initializer = tf.keras.initializers.
 ↪he_normal(seed=42),kernel_regularizer=tf.keras.regularizers.l1())(max_pool_2)
```

```
max_pool_3 = MaxPool1D(5)(i1)

flatten = Flatten()(max_pool_3)

dropout_layer = Dropout(0.5)(flatten)

dense_layer1 = Dense(256,activation="relu",kernel_initializer = tf.keras.
 ↪initializers.he_normal(seed=42))(dropout_layer)



output_layer = Dense(20,activation="softmax",kernel_initializer= tf.keras.
 ↪initializers.glorot_normal(seed=42))(dense_layer1)


model =Model(inputs=first_layer,outputs=output_layer)
```

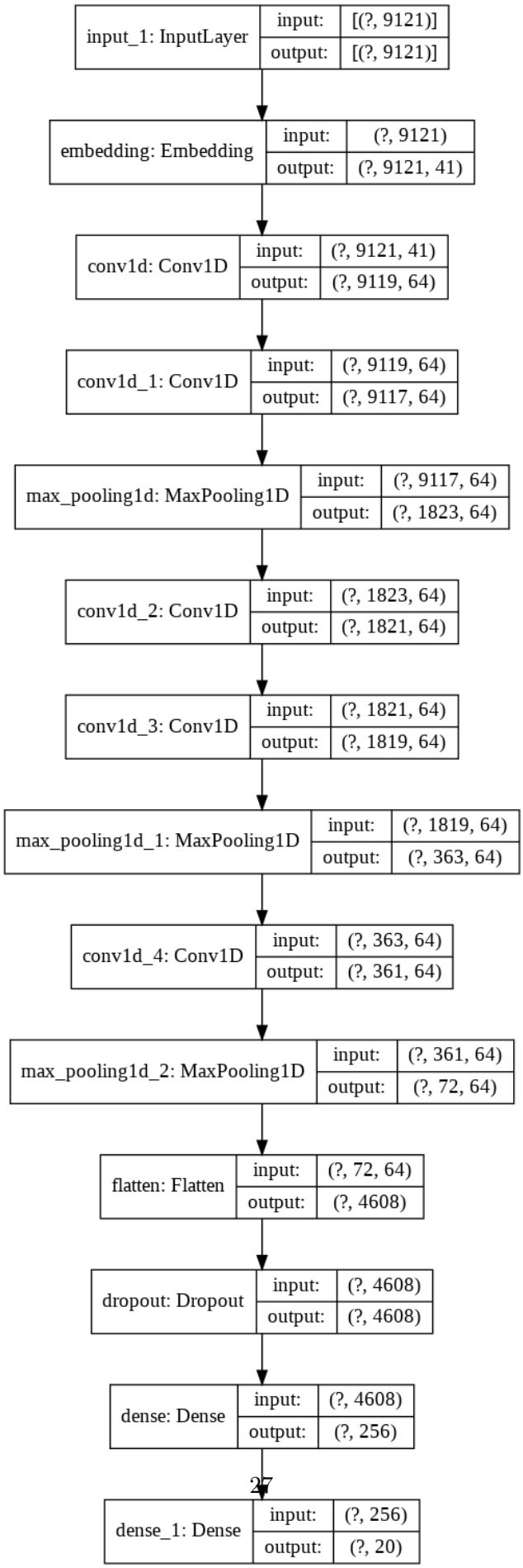
[40]:
```
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 9121)]            0
_________________________________________________________________
embedding (Embedding)        (None, 9121, 41)          1681
_________________________________________________________________
conv1d (Conv1D)              (None, 9119, 64)          7936
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 9117, 64)          12352
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 1823, 64)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 1821, 64)          12352
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 1819, 64)          12352
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 363, 64)           0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 361, 64)           12352
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 72, 64)            0
_________________________________________________________________
flatten (Flatten)            (None, 4608)              0
_________________________________________________________________
dropout (Dropout)            (None, 4608)              0
```

```
_________________________________________________________________
dense (Dense)                (None, 256)               1179904
_________________________________________________________________
dense_1 (Dense)              (None, 20)                5140
=================================================================
Total params: 1,244,069
Trainable params: 1,242,388
Non-trainable params: 1,681
_________________________________________________________________
```

[41]:
```
tf.keras.utils.plot_model(model,to_file = 'model2.
↪png',show_shapes=True,show_layer_names=True)
```

[41]:


input_1: InputLayer
input: [(?, 9121)]
output: [(?, 9121)]
embedding: Embedding
input: (?, 9121)
output: (?, 9121, 41)
conv1d: Conv1D
input: (?, 9121, 41)
output: (?, 9119, 64)
conv1d_1: Conv1D
input: (?, 9119, 64)
output: (?, 9117, 64)
max_pooling1d: MaxPooling1D
input: (?, 9117, 64)
output: (?, 1823, 64)
conv1d_2: Conv1D
input: (?, 1823, 64)
output: (?, 1821, 64)
conv1d_3: Conv1D
input: (?, 1821, 64)
output: (?, 1819, 64)
max_pooling1d_1: MaxPooling1D
input: (?, 1819, 64)
output: (?, 363, 64)
conv1d_4: Conv1D
input: (?, 363, 64)
output: (?, 361, 64)
max_pooling1d_2: MaxPooling1D
input: (?, 361, 64)
output: (?, 72, 64)
flatten: Flatten
input: (?, 72, 64)
output: (?, 4608)
dropout: Dropout
input: (?, 4608)
output: (?, 4608)
dense: Dense
input: (?, 4608)
output: (?, 256)
dense_1: Dense
input: (?, 256)
output: (?, 20)

[42]:
```
%load_ext tensorboard
```

```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
```

[43]:
```
from sklearn.metrics import f1_score
class custom(tf.keras.callbacks.Callback):

  def on_train_begin(self,logs={}):
    self.f1_score_list = []

  def on_epoch_end(self,epoch,logs={}):

    x_val,y_val = x_test,y_test

    pred_y = self.model.predict(x_val)

    y_t = np.zeros(y_val.shape[0])
    y_p = np.zeros(pred_y.shape[0])

    for i in range(len(y_t)):
      y_t[i] = int(np.argmax(y_val[i]))
      y_p[i] = int(np.argmax(y_p[i]))

    f1_value = f1_score(y_t,y_p,average="micro")
    print("f1_score:",f1_value)

    self.f1_score_list.append(f1_value)
```

[44]:
```
f1_call = custom()

## Callback for saving best model
checkpoint = ModelCheckpoint(filepath='best_model_1.
 ↪h5',verbose=1,monitor='val_accuracy',
                              mode='auto',save_best_only=True)

## Callback for earlystopping
early_stop =␣
 ↪EarlyStopping(monitor="val_accuracy",mode='max',patience=2,verbose=1)

## Tensorboard
log_dir = "logs"
tensorboard = TensorBoard(log_dir=log_dir,histogram_freq=1,write_graph=True)
## reducing learning rate
```

```
reduce_ = tf.keras.callbacks.
 ↪ReduceLROnPlateau(monitor='val_accuracy',patience=1,mode='auto',verbose=1,factor=0.
 ↪9)

## all callbacks
callbacks =[reduce_ , f1_call,checkpoint,early_stop,tensorboard]

## compile model
model.compile(loss='categorical_crossentropy', optimizer=tf.keras.optimizers.
 ↪Adam(learning_rate=0.001), metrics=['accuracy'])


## Trainning


model.fit(x_train,y_train,epochs=15,validation_data=(x_test,y_test),batch_size␣
 ↪=64,callbacks=callbacks)
```

```
Epoch 1/15
  2/221 […] - ETA: 1:25 - loss: 52.6266 - accuracy:
0.0547WARNING:tensorflow:Method (on_train_batch_end) is slow compared to the
batch update (0.277517). Check your callbacks.
221/221 [==============================] - ETA: 0s - loss: 18.1777 - accuracy:
0.0644f1_score: 0.04248990864669641

Epoch 00001: val_accuracy improved from -inf to 0.05269, saving model to
best_model_1.h5
221/221 [==============================] - 65s 296ms/step - loss: 18.1777 -
accuracy: 0.0644 - val_loss: 3.4179 - val_accuracy: 0.0527 - lr: 0.0010
Epoch 2/15
221/221 [==============================] - ETA: 0s - loss: 3.0929 - accuracy:
0.0477
Epoch 00002: ReduceLROnPlateau reducing learning rate to 0.0009000000427477062.
f1_score: 0.04248990864669641

Epoch 00002: val_accuracy did not improve from 0.05269
221/221 [==============================] - 62s 281ms/step - loss: 3.0929 -
accuracy: 0.0477 - val_loss: 3.0580 - val_accuracy: 0.0523 - lr: 0.0010
Epoch 3/15
221/221 [==============================] - ETA: 0s - loss: 3.0502 - accuracy:
0.0511f1_score: 0.04248990864669641

Epoch 00003: val_accuracy improved from 0.05269 to 0.05290, saving model to
best_model_1.h5
221/221 [==============================] - 63s 283ms/step - loss: 3.0502 -
accuracy: 0.0511 - val_loss: 3.0490 - val_accuracy: 0.0529 - lr: 9.0000e-04
Epoch 4/15
```

```
221/221 [==============================] - ETA: 0s - loss: 3.0496 - accuracy:
0.0496
Epoch 00004: ReduceLROnPlateau reducing learning rate to 0.0008100000384729356.
f1_score: 0.04248990864669641

Epoch 00004: val_accuracy did not improve from 0.05290
221/221 [==============================] - 62s 282ms/step - loss: 3.0496 -
accuracy: 0.0496 - val_loss: 3.0486 - val_accuracy: 0.0527 - lr: 9.0000e-04
Epoch 5/15
221/221 [==============================] - ETA: 0s - loss: 3.0423 - accuracy:
0.0476
Epoch 00005: ReduceLROnPlateau reducing learning rate to 0.0007290000503417104.
f1_score: 0.04248990864669641

Epoch 00005: val_accuracy did not improve from 0.05290
221/221 [==============================] - 62s 281ms/step - loss: 3.0423 -
accuracy: 0.0476 - val_loss: 3.0422 - val_accuracy: 0.0525 - lr: 8.1000e-04
Epoch 00005: early stopping
```

[44]:
```
<tensorflow.python.keras.callbacks.History at 0x7f4320449160>
```

[45]:
```
%tensorboard --logdir logs
```

```
Output hidden; open in https://colab.research.google.com to view.
```