**Ngram Model**

*# code courtesy of https://nlpforhackers.io/language-models/*

**from** **nltk.corpus** **import** reuters

*# from nltk import bigrams, trigrams*

**from** **nltk** **import** ngrams

**from** **collections** **import** Counter, defaultdict

**import** **itertools**

**def** getAlarmsSeqs(fname):

lines = **None**

**with** open(fname) **as** f:

lines = f.readlines()

temp\_alarms\_seqs = []

**for** i,line **in** enumerate(lines):

seq = line.split()

temp\_alarms\_seqs.append(seq)

temp\_alarms\_seqs = list(itertools.chain(\*temp\_alarms\_seqs))

**return** temp\_alarms\_seqs

fname = "../.data/train.tokens"

train\_alarms\_seqs = getAlarmsSeqs(fname)

*# train\_alarms\_seqs[0:50]*

*# Create a placeholder for model*

model = defaultdict(**lambda**: defaultdict(**lambda**: 0))

n = 12

**for** tup **in** ngrams(train\_alarms\_seqs, n=n, pad\_right=**True**, pad\_left=**True**):

*# print(type(tup[0:n-1]),len(tup[0:n-1]))*

*# print(a1,a2,a3,a4,a5)*

model[tup[0:n-1]][tup[-1]] += 1

*# Let's transform the counts to probabilities*

**for** key **in** model:

*# print(key)*

total\_count = float(sum(model[key].values()))

**for** w3 **in** model[key]:

model[key][w3] /= total\_count

*# d = dict(model[ '47TI2086',*

*# '47PDI003',*

*# '47TI3418',*

*# '47LI003',*

*# '47LI003'])*

*# print(list(d.keys())[0],list(d.values())[0])*

*# print(d)*

*# calculate the accuracy*

fname = "../.data/val.tokens"

val\_alarms\_seqs = getAlarmsSeqs(fname)

alarm2acc = {}

**for** a **in** train\_alarms\_seqs:

alarm2acc[a] = {"true":0, "false":0, "acc":0}

**for** a **in** val\_alarms\_seqs:

alarm2acc[a] = {"true":0, "false":0, "acc":0}

**for** i **in** range(len(val\_alarms\_seqs)-n):

seq = train\_alarms\_seqs[i:i+n]

*# print("seq =>",seq)*

in\_seq = seq[0:n-1]

y\_true = seq[-1]

*# print(in\_seq,y\_true)*

*# print(seq\_in,y\_true)*

d = dict(model[tuple(in\_seq)])

y\_pred = list(d.keys())[0]

**if** y\_pred == y\_true:

alarm2acc[y\_true]['true'] += 1

**else**:

alarm2acc[y\_true]['false'] += 1

*# print(list(d.keys())[0],list(d.values())[0])*

**for** a **in** alarm2acc.keys():

**if** alarm2acc[a]["true"]+alarm2acc[a]["false"]>0:

alarm2acc[a]["acc"] =alarm2acc[a]["true"] /(alarm2acc[a]["true"]+alarm2acc[a]["false"])

more\_than\_50\_percent\_acc = [(a,alarm2acc[a]["acc"]) **for** a **in** alarm2acc.keys() **if** alarm2acc[a]["acc"]>=0.5]

print(f"Number of Alarm Sources with more than 50% Accuracy = **{**len(more\_than\_50\_percent\_acc)**}**")

*# alarm2acc*

**for** x **in** [0.5,0.6,0.7,0.8,0.9]:

more\_than\_x\_percent\_acc = [(a,alarm2acc[a]["acc"]) **for** a **in** alarm2acc.keys() **if** alarm2acc[a]["acc"]>=x]

print(f"Number of Alarm Sources with more than **{**x**}**% Accuracy = **{**len(more\_than\_x\_percent\_acc)**}**")