HU Extension Assignment 08 E63 Big Data Analytics

Handed out: 10/20/2017 Due by 4:00 PM EST on Saturday, 10/28/2017

If you are familiar with NLP API-s in languages other than Python or Python NLP API-s other than NLTK please be free to solve these problems using technology of your choice.

Problem 1. Use the text of the Universal Declaration of Human Rights (UDHR). Create a table for 5 languages in which you will collect statistics about the languages used. Place in that table the number of words in each language in UDHR, number of unique words, average length of words, number of sentences contained in UDHR and average number of words per sentence. Create a distribution of sentence lengths for each language. Plot those (non-cumulative) distributions on one diagram.
(25%)

See p1.ipyb for a closer look

1. Use the text of the Universal Declaration of Human Rights (UDHR).

In [33]:
```
udhr.fileids()
```

Out[33]:
```
[u'Abkhaz-Cyrillic+Abkh',
 u'Abkhaz-UTF8',
 u'Achehnese-Latin1',
 u'Achuar-Shiwiar-Latin1',
 u'Adja-UTF8',
 u'Afaan_Oromo_Oromiffa-Latin1',
 u'Afrikaans-Latin1',
 u'Aguaruna-Latin1',
 u'Akuapem_Twi-UTF8',
 u'Albanian_Shqip-Latin1',
 u'Amahuaca',
 u'Amahuaca-Latin1',
 u'Amarakaeri-Latin1',
 u'Amuesha-Yanesha-UTF8',
 u'Arabela-Latin1',
 u'Arabic_Alarabia-Arabic',
 u'Asante-UTF8',
 u'Ashaninca-Latin1',
 u'Asheninca-Latin1',
```

2. Create a table for 5 languages in which you will collect statistics about the languages used.

```
languages = [
                    'English-Latin1',
                    'Danish_Dansk-Latin1',
                    'German_Deutsch-Latin1',
                    'Filipino_Tagalog-Latin1',
                    'Italian-Latin1'
              ]
```

3. Place in that table the number of words in each language in UDHR, number of unique words, average length of words, number of sentences contained in UDHR and average number of words per sentence.

```
##https://stackoverflow.com/questions/35900029/average-sentence-length-for-every-text-in-corpus-python3-nltk

np_object=[]
for lang in languages:
    chars_count = len(udhr.raw(lang))
    word_count = len(udhr.words(lang))
    unique_word_count = len(set(udhr.words(lang)))
    word_length_avg = round(chars_count/word_count)
    #word_length_avg = sum(len(sent) for sent in udhr.sents(fileids=[lang])) / len(udhr.sents(fileids=[lang]))
    sents_count = len(udhr.sents(lang))
    avg_num_words_per_sents = round(word_count/sents_count)
    x = [
            lang,
            word_count,
            unique_word_count,
            word_length_avg,
            sents_count,
            avg_num_words_per_sents
        ]
    np_object.append(x)
    x = []
df = pd.DataFrame(np_object, columns=['language', 'word_count', 'word_count_unique', 'word_length_avg', 'sents_count',
df
```

	language	word_count	word_count_unique	word_length_avg	sents_count	avg_num_words_per_sents
0	English-Latin1	1781	533	5.0	67	26.0
1	Danish_Dansk-Latin1	1696	584	5.0	86	19.0
2	German_Deutsch-Latin1	1521	579	6.0	60	25.0
3	Filipino_Tagalog-Latin1	1803	480	5.0	75	24.0
4	Italian-Latin1	1723	578	5.0	51	33.0

4. Create a distribution of sentence lengths for each language.

```
cfd = nltk.ConditionalFreqDist(
            (lang, len(sent))
            for lang in languages
            for sent in udhr.sents(lang))

# (lang, (sum(len(sent))/ len(udhr.sents(fileids=[lang])))
cfd
```

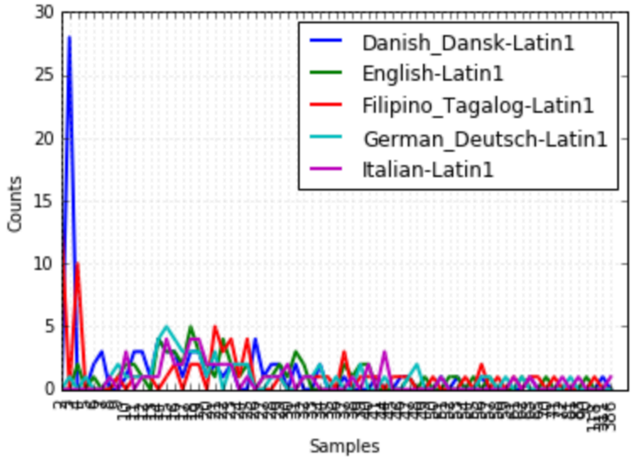
```
ConditionalFreqDist(nltk.probability.FreqDist,
                    {'Danish_Dansk-Latin1': FreqDist({3: 28,
                              6: 2,
                              7: 3,
                              10: 1,
                              11: 3,
                              12: 3,
                              13: 1,
                              14: 4,
                              15: 3,
                              16: 3,
                              17: 1,
                              18: 3,
                              19: 3,
                              20: 1,
                              21: 2,
                              22: 1,
                              23: 2,
                              26: 4,
```

```
cfd.tabulate(conditions=languages, samples=range(10), cumulative=True)
```

```
                          0  1  2  3  4  5  6  7  8  9
         English-Latin1   0  0  0  0  2  2  3  3  3  4
    Danish_Dansk-Latin1   0  0  0 28 28 28 30 33 33 33
  German_Deutsch-Latin1   0  0  0  1  1  2  2  2  3  5
Filipino_Tagalog-Latin1   0  0 13 13 23 23 23 23 23 24
         Italian-Latin1   0  0  0  0  0  0  0  0  1  1
```

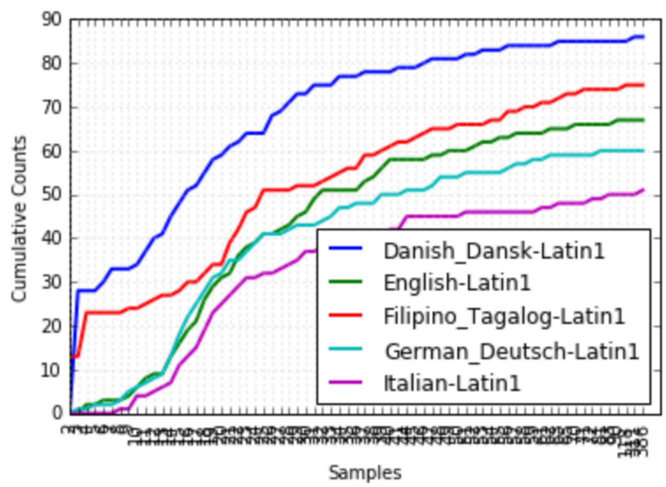
5. Plot those (non-cumulative) distributions on one diagram.

```
cfd.plot(cumulative=False)
```



Plotting for fun cumualtive distributions on one diagram

```
]: cfd.plot(cumulative=True)
```



Problem 2. Identify 10 most frequently used words longer than 7 characters in the entire corpus of Inaugural Addresses. Do not identify 10 words for every speech but rather 10 words for the entire corpus. Which among those words has the largest number of synonyms? List all

synonyms for those 10 words. Which one of those 10 words has the largest number of hyponyms? List all hyponyms of those 10 most frequently used "long" words. The purpose of this problem is to familiarize you with WordNet and concepts of synonyms and hyponyms.
(25%)
Your literature for Problems 1 and 2 are chapters 1 and 2 of Natural Language Processing with Python book by Steven Bird et al.

See p2.ipyb for a closer look

1. Identify 10 most frequently used words longer than 7 characters in the entire corpus of Inaugural Addresse

- Do not identify 10 words for every speech but rather 10 words for the entire corpus.

```
word_extract=[word.lower() for word in inaugural_corpus if (len(word) > 7 and word.isalpha())]
word_frequency=FreqDist(word_extract)
print("\n Top 10 words in Inauguaral Corpus-- ignores case and non-alpha chars")
word_frequency.tabulate(10)
print word_freq
```

```
 Top 10 words in Inauguaral Corpus-- ignores case and non-alpha chars
  government     citizens constitution     national     american     congress    interests    political    execu
  principles
         593          237          205          154          147          129          113          106
          93
None
```

2. Which among those words has the largest number of synonyms?

- List all synonyms for those 10 words.

```
# wn.synsets('motorcar')
top_10_words = [
                    'government',
                    'citizens',
                    'constitution',
                    'national',
                    'american',
                    'congress',
                    'interests',
                    'political',
                    'executive',
                    'principles'
             ]
syn_words = {}
for word in top_10_words:
    syn_words[word] = []
    syn = wn.synsets(word)
    for s in syn:
        lemmas = s.lemma_names()
        if len(lemmas) > 0:
            for l in lemmas:
                l = l.lower()
                if l is not word and l.encode('utf-8') not in syn_words[word]:
                    syn_words[word].append(l.encode('utf-8'))
pprint.pprint(syn_words)

# pprint.pprint([i for synset in wn.synsets(word) for i in synset.lemma_names()])
```

```
{'american': ['american', 'american_english', 'american_language'],
 'citizens': ['citizen'],
 'congress': ['congress',
              'united_states_congress',
              'u.s._congress',
              'us_congress',
              'sexual_intercourse',
              'intercourse',
              'sex_act',
              'copulation',
              'coitus',
              'coition',
              'sexual_congress',
              'sexual_relation',
              'relation',
              'carnal_knowledge'],
 'constitution': ['fundamental_law',
                  'organic_law',
                  'constitution',
                  'establishment',
                  'formation',
                  'organization',
                  'organisation',
                  'united_states_constitution',
                  'u.s._constitution',
                  'us_constitution',
                  'constitution_of_the_united_states',
                  'composition',
                  'physical_composition',
                  'makeup',
                  'make-up',
                  'old_ironsides'],
```

We can see that constitution is the top 10 word with the most synonyms

```
for word in syn_words:
    print(str(word)+":" +str(len(syn_words[word])))
```

```
interests:12
executive:3
constitution:17
congress:15
government:9
national:5
citizens:1
political:1
principles:4
american:3
```

3. List all hyponyms of those 10 most frequently used “long” words. ¶

- Which one of those 10 words has the largest number of hyponyms?
- The purpose of this problem is to familiarize you with WordNet and concepts of synonyms and hyponyms.

]:
```
top_10_words = [
                    'government',
                    'citizens',
                    'constitution',
                    'national',
                    'american',
                    'congress',
                    'interests',
                    'political',
                    'executive',
                    'principles'
             ]
hyponym_words = {}
for word in top_10_words:
    hyponym_words[word] = []
    syn = wn.synsets(word)
    for s in syn:
        hypos = s.hyponyms()
        if len(hypos) > 0:
            for hyp in hypos:
                for h in hyp.lemma_names():
                    h = h.lower()
                    if h is not word and h.encode('utf-8') not in hyponym_words[word]:
                        hyponym_words[word].append(h.encode('utf-8'))
pprint.pprint(hyponym_words)
```

```
{'american': ['african-american',
              'african_american',
              'afro-american',
              'black_american',
              'alabaman',
              'alabamian',
              'alaskan',
              'anglo-american',
              'appalachian',
              'arizonan',
              'arizonian',
              'arkansan',
              'arkansawyer',
              'asian_american',
              'bay_stater',
              'bostonian',
              'californian',
              'carolinian',
              'coloradan',
```

We can see that american is the top 10 word with the most hyponyms

```
for word in hyponym_words:
    print(str(word)+":" +str(len(hyponym_words[word])))
```

```
interests:43
executive:18
constitution:17
congress:18
government:32
national:4
citizens:9
political:0
principles:62
american:109
```

Problem 3. Create your own grammar for the following sentence: "Describe every step of your work and present all intermediate and final results in a Word document".
(10%)
Your literature for Problem 3 is chapter 8 of Natural Language Processing with Python book by Steven Bird et al.

See p3.ipyb for a closer look

```
In [1]: import nltk
```

Problem 3

1. Create your own grammar for the following sentence:
   - "Describe every step of your work and present all intermediate and final results in a Word document".

```
In [2]: sentence = """Describe every step of your work and present all intermediate and final results in a Word dc
```

```
In [3]: tokens = nltk.word_tokenize(sentence)
```

```
In [4]: tokens
```

```
Out[4]: ['Describe',
         'every',
         'step',
         'of',
         'your',
         'work',
         'and',
         'present',
         'all',
         'intermediate',
         'and',
         'final',
         'results',
         'in',
         'a',
         'Word',
         'document']
```

```
In [5]: tagged_text = nltk.pos_tag(tokens)
```

```
In [6]: tagged_text
```

```
Out[6]: [('Describe', 'NNP'),
         ('every', 'DT'),
         ('step', 'NN'),
         ('of', 'IN'),
         ('your', 'PRP$'),
         ('work', 'NN'),
         ('and', 'CC'),
         ('present', 'JJ'),
         ('all', 'DT'),
         ('intermediate', 'JJ'),
         ('and', 'CC'),
         ('final', 'JJ'),
         ('results', 'NNS'),
         ('in', 'IN'),
         ('a', 'DT'),
         ('Word', 'NNP'),
         ('document', 'NN')]
```

Problem 4. Install and compile Word2Vec C executables. Train CBOW model and create 200 dimensional embedding of Word Vectors. Demonstrate that you could run analogical reasoning when searching for country's favorite food starting with japan and sushi. Note that words might have to be in lower case. Find favorite food for 5 different countries. Report improbable results as well as good results. Use scripts provided with original Google C code.
(20%)

```
Enter word or sentence (EXIT to break): EXIT
swaite@Rmt-mac-swaite:~/stirling/CSIE-63/assignment-8/word2vec|master⚡
⇒  sudo ./demo-analogy.sh
make: Nothing to be done for `all'.
-----------------------------------------------------------------------------------------------
Note that for the word analogy to perform well, the models should be trained on much larger data set
Example input: paris france berlin
-----------------------------------------------------------------------------------------------
Starting training using file text8
Vocab size: 71290
Words in train file: 16718843
Alpha: 0.000123  Progress: 99.57%  Words/thread/sec: 73.12k
real    0m43.345s
user    3m59.424s
sys     0m1.064s
```

japan sushi thailand

```
Enter three words (EXIT to break): japan sushi thailand

Word: japan  Position in vocabulary: 582

Word: sushi  Position in vocabulary: 30679

Word: thailand  Position in vocabulary: 5640

                                              Word
Distance
------------------------------------------------------------------
                                              crab      0.561080
                                             bento      0.559273
                                            kimchi      0.541000
                                             mochi      0.540748
                                             crepe      0.536611
```

japan sushi italy

```
Enter three words (EXIT to break): japan sushi italy

Word: japan  Position in vocabulary: 582

Word: sushi  Position in vocabulary: 30679

Word: italy  Position in vocabulary: 843

                                              Word
Distance
------------------------------------------------------------------
                                      strawberries      0.500201
                                              kelp      0.500120
                                           sprouts      0.495927
                                          omelette      0.493959
                                            mussel      0.491622
                                              pies      0.489951
                                       cranberries      0.474655
```

japan sushi mexico

```
Enter three words (EXIT to break): japan sushi mexico

Word: japan  Position in vocabulary: 582

Word: sushi  Position in vocabulary: 30679

Word: mexico  Position in vocabulary: 1352

                                              Word              Distance
------------------------------------------------------------------------
                                         barbecued              0.556577
                                            manioc              0.536024
                                              tofu              0.514035
                                        perforated              0.512456
                                              kare              0.512113
                                             soups              0.510998
                                          leftover              0.506907
                                              thud              0.506563
                                             feeds              0.506198
                                      cookbookwiki              0.503084
                                            crispy              0.498741
                                         boutiques              0.497999
                                         subgenera              0.497614
                                            leaved              0.496927
                                         marinated              0.496811
                                      grasshoppers              0.495767
                                             snack              0.495525
                                         glutinous              0.495342
                                        seasonings              0.495112
                                         chrysalis              0.494900
```

japan sushi canada

```
Enter three words (EXIT to break): japan sushi canada

Word: japan  Position in vocabulary: 582

Word: sushi  Position in vocabulary: 30679

Word: canada  Position in vocabulary: 474

                                                    Word              Distance
------------------------------------------------------------------------
                                                    kare              0.616777
                                               glutinous              0.540952
                                            cookbookwiki              0.535959
                                                   thyme              0.534985
                                               boutiques              0.530151
                                                   hives              0.530043
                                                lamiales              0.523256
                                             caterpillar              0.521952
                                                 juniper              0.519933
                                                  cherry              0.518585
                                                    allo              0.518086
                                               buckwheat              0.516296
                                                 sprouts              0.514408
                                                 surfing              0.513681
                                              bullwinkle              0.512200
                                                leftover              0.510922
                                                   bento              0.510452
                                                sapphire              0.509470
                                                 pickles              0.509440
                                                  mussel              0.508209
                                                molluscs              0.506993
                                               roundworm              0.506693
                                                 parsley              0.505978
```

japan sushi india

```
Enter three words (EXIT to break): japan sushi india

Word: japan  Position in vocabulary: 582

Word: sushi  Position in vocabulary: 30679

Word: india  Position in vocabulary: 508

                                                 Word              Distance
------------------------------------------------------------------------
                                          cookbookwiki             0.534107
                                                manioc             0.525004
                                               juniper             0.514114
                                                  weed             0.513496
                                                crispy             0.511961
                                                 mochi             0.511274
                                             glutinous             0.509846
                                               breaded             0.508894
                                                 drawl             0.506393
                                              leftover             0.506105
                                               sprouts             0.504681
                                                shakin             0.504605
                                             thumbnail             0.503094
                                               cayenne             0.499609
                                                   lha             0.497450
                                                  soya             0.495425
                                               parsley             0.495213
                                                  thud             0.494816
                                              lamiales             0.494529
                                                  kare             0.494128
                                                  ebay             0.493733
```

Problem 5. Install and run Genism Python Word2Vec API. Find the most probable words you will obtain when you start with an emperor add a woman and subtract a man. Use this tutorial as a guide https://rare-technologies.com/word2vec-tutorial/ (20%)

See p5.ipyb for a closer look

In [1]:
```
# import modules & set up logging
import gensim, logging
```

Working on a Macbook Pro so needed to use the link below to get this code to work properly

- https://github.com/William-Yeh/word2vec-mac

In [6]:
```
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
```

In [7]:
```
#load model, model was created in problem 1
model = gensim.models.KeyedVectors.load_word2vec_format('./word2vec/vectors.bin', binary=True)
```

```
2017-10-28 01:43:49,221 : INFO : loading projection weights from ./word2vec/vectors.bin
2017-10-28 01:43:50,188 : INFO : loaded (71290, 200) matrix from ./word2vec/vectors.bin
```

In [8]:
```
#what requested in the problem
model.most_similar(positive=['emperor', 'woman'], negative=['man'], topn=1)
```

```
2017-10-28 01:43:50,236 : INFO : precomputing L2-norms of word weight vectors
```

Out[8]:
```
[(u'montoku', 0.6238285303115845)]
```

Analysis of output

These results look correct. Looking up "montoku" on wikipedia gave the following results. https://en.wikipedia.org/wiki/Emperor_Montoku

"Emperor Montoku was the 55th emperor of Japan, according to the traditional order of succession. Montoku's reign lasted from 850 to 858."

Please, describe every step of your work and present all intermediate and final results in a Word document. Please, copy past text version of all essential command and snippets of results into the Word document with explanations of the purpose of those commands. We cannot retype text that is in JPG images. Please, always submit a separate copy of the original, working scripts and/or class files you used. Sometimes we need to run your code and retyping is too costly. Please include in your MS Word document only relevant portions of the console output or output files. Sometime either console output or the result file is too long and including it into the MS Word document makes that document too hard to read. PLEASE DO NOT EMBED files into your MS Word document. For issues and comments visit the class Discussion Board. If you use some other language other than Python in your daily work with NLP, please be free to use that language and a framework of your choice to do this assignment.