S2-SantiagoBernal-PereAyats2

September 19, 2018

In [3]:
```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /usr/local/share/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[3]: True

In [6]:
```
from nltk.metrics import jaccard_distance
from scipy.stats import pearsonr
input_file = 'trial/STS.input.txt'
with open(input_file) as f:
    input_data = f.readlines()
distances = []
for i in input_data:
    sentences = nltk.sent_tokenize(i[4:])
    words = [nltk.word_tokenize(sent) for sent in sentences]
    distances.append(jaccard_distance(set(words[0]),set(words[1])))
print(distances)
```

```
[0.6923076923076923, 0.7368421052631579, 0.6666666666666666, 0.5454545454545454, 0.769230769230
```

In [8]:
```
gold_file = 'trial/STS.gs.txt'
with open(gold_file) as f:
    gold_data = f.readlines()
gold = [int(g[4:5]) for g in gold_data]
pearsonr(distances, gold)[0]

# We think that it performs quite well considering
# the amount of information given since as the pearson
# correlation comes closer to 1, we get closer and closer
# to reach a total positive linear correlation.
# With a score of 0.414, the similarities calculated from the
# input data increased accordingly with the gold standard
# with some degree.This considering as well that jaccard
# distance is very sensitive with small sample sizes and
```

```
        # can give very erronous results when small data samples
        # are use, as is the case.
```

Out[8]: 0.4143770872333895

In [8]:
```
        # Optional exercise
        from nltk.collocations import TrigramCollocationFinder
        from nltk.metrics import jaccard_distance, ConfusionMatrix
        from nltk.metrics.scores import accuracy
        import numpy as np


        def get_trigrams(lang_text, filt):
            finder = TrigramCollocationFinder.from_words(lang_text)
            finder.apply_freq_filter(filt)
            #trigrams = sorted([tr for tr in finder.ngram_fd.items()], key=lambda x: x[1], rev
            trigrams = [tr for tr in finder.ngram_fd.items()]
            return [x[0] for x in trigrams]

        files = ["eng", "spa", "nld", "deu", "ita", "fra"]

        # Precalc training trigrams of each language
        train_trigrams = []
        for f in files:
            with open("langId/" + f + "_trn.txt") as lang_file:
                lf = lang_file.readlines()
            lang_text = "  ".join([" ".join([x for w in l.split() for x in ("".join(c for c in
            train_trigrams.append(get_trigrams(lang_text, 3000))

        # Test trigrams for each sentence of each language
        tests = []
        test_trigrams = []
        for f in files:
            with open("langId/" + f + "_tst.txt") as lang_file:
                lf = lang_file.readlines()
            lang_sentences = [" ".join([x for w in l.split() for x in ("".join(c for c in w if
            lang_trigram=[]
            for sentence in lang_sentences:
                lang_trigram.append(get_trigrams(sentence,0))
            test_trigrams.append(lang_trigram)
```

In [9]:
```
        # The distance of each test sentence to each language
        # grouped by language to validate with the labels later
        distances = []
        for lang in test_trigrams:
            distances_lang = []
            for sentence in lang:
                distances_sent = []
```

```
            for tt in train_trigrams:
                distances_sent.append(jaccard_distance(set(tt), set(sentence)))
            distances_lang.append(distances_sent)
        distances.append(distances_lang)

    # gets the expected label (refs) and real label (results) of the tests
    refs = []
    count = 0
    results = []
    for i in distances:
        for c in i:
            results.append(files[c.index(min(c))])
            refs.append(files[count])
        count +=1
```

In [10]:
```
# Display information
'accuracy: ' + str(accuracy(refs,results))

# The accuracy is pretty high and works well.
# To be able to obtain this accuracy,
# the amount of fequency filter used for the training
# trigram had to be increased so that the jaccard
# distance was able to function better. We think
# this is because since the jaccard distance doesnt
# take into consideration the frequency of the trigram
# when calculating a distance but rather the membership
# of the trigram on both sets vs the non-membership,
# sometimes the distance
# of a sentence was closer to a wrong language simply
# because that language maybe had a lot of the trigrams
# of the sentence even if the trigrams weren't very frequent
# at all in the language. Fortunately, by setting a higher
# threshold of the amount of repetitions a trigram needs in
# order to consider it for the distance removes this problem
# and the distance is able to be calculated more accurately.
```

Out[10]:
```
'accuracy: 0.9873618220317788'
```

In [11]:
```
cm = ConfusionMatrix(refs,results)
print(cm.pretty_format())

# Viewing the Confusion Matrix, some of the mistakes
# of the language identifier makes sense, considering
# that the errors seem to be mostly on languages that
# are similar or share common words. For example,
# for german, dutch is similar and most of the
# mistakes made identifying german where made because
# the project considered it as dutch and viceversa,
```

```
        # most of the mistakes in dutch was because they where
        # falsesly identified as german. German was
        # also confused as english in some cases, which also
        # makes sense since they share some words and english
        # has a lot of words that have their origin in the
        # german language. Another example is with italian
        # and french being confused with spanish, which makes
        # sense as they all come from the Italo-Western branch
        # of the Latin language.
```

```
    |    d    e    f    i    n    s |
    |    e    n    r    t    l    p |
    |    u    g    a    a    d    a |
----+-------------------------------+
deu |<9867>  26    9    2   80    6 |
eng |   30<9872>  42   11   23    9 |
fra |   23   23<9884>  19   19   32 |
ita |    4   22   24<9822>   4  124 |
nld |   55   38   13    4<9883>   7 |
spa |   15   17   31   42    4<9891>|
----+-------------------------------+
(row = reference; col = test)
```