

Assignment 3 FOR DUMMIES

So we got NGram corpus with a lot of sentences.

Each sentence format is:

```
head_word<TAB>syntactic-gram<TAB>total_count<TAB>counts_by_year
```

and the syntatic-gram format is:

The syntactic-gram format is a space-separated list of tokens, each token format is "word/pos-tag/dep-label/head-index".

so we got tokens,

each token has 'word' (like 'likes'), 'dep-label' (like 'subject'), 'index' for the next word related to it (like 'dog').

So we will get from it "word1 word2 dep-label" and add to it "total count"

So step 1 we will get all syntatic-gram inputs, and their total count.

and output "word1 word2 dep-label total-count" for mapper

this is after we used stemmer on the words.

(stemmer takes word like "walked" | "walking" and change them to "walk")

and step 1 reducer will combine them (see example in the end).



So there are all kinds of dep-label. So in Step 1A we collect all of the used types of dep-labels to use in step 2.

step 1A output can be like "comp det obj prep ..."
(dep types used names)

In step 2 we create for each words pair a vector.

The vector is the size of the amount of dep-labels types.

So if in step 1A we got 5 types, each coordinate will be the amount of appearances of that specific dep-labels.

"word1 word2 0 0 25 3 8"

Mapper put all the types to the relevant key
reducer created the vectors.

Example:

Step 1 input (ass3inputtemp.txt):

```
App.java  Step2.java  ass3inputtemp.txt x
cease<tab>walked/VB/ccomp/0 for/IN/prep/1 an/DT/det/4 instant/NN/pobj/2<tab>56<tab>1834,2 1835,1 1856,1 1863,1 1871,1 1872,1
cease<tab>cease/VB/ccomp/0 for/IN/prep/1 an/DT/det/4 boys/NN/pobj/2<tab>56<tab>1834,2 1835,1 1856,1 1863,1 1871,1 1872,1 1
cease<tab>cease/VB/ccomp/0 for/IN/prep/1 an/DT/det/4 boys/NN/pobj/2<tab>56<tab>1834,2 1835,1 1856,1 1863,1 1871,1 1872,1 1
```

output:

```
an boi det 112
an instant det 56
boi for pobj 112
ceas root ccomp 112
for ceas prep 112
for walk prep 56
walk root ccomp 56
ps c:\Users\pavet\Bart
```

step 1A input

step 1A output:

```
ccomp
det
pobj
prep
ps c:\Users
```

(all dep-types)

(will be our vector)

step 2 input

step 2 output:

```
an,boi 0 112 0 0
an,instant 0 56 0 0
boi,for 0 0 112 0
ceas,root 112 0 0 0
for,ceas 0 0 0 112
for,walk 0 0 0 56
instant,for 0 0 56 0
walk,root 56 0 0 0
ps c:\Users
```

our vectors!

!!

WHAT NOW? (not sure)

I've added Equations.java with the requested equations.

I think we need to create new vectors with the calculations of the step 2 equations.

like

"word1 word2 L1 L2 Cosine Jaccard Pice Jensen" 1×6

somehow. P or do it in the classifier?

After that using the word-relatedness.txt and the other 4 requested equations.

To train a model with sized 24 matrix

	e1	e2	e3	e4	e5	e6
equation 1	X	X	X	X	X	X
e2	X	X	X	X	X	X
e3	X	X	X	X	X	X
e4	X	X	X	X	X	X

(size 24)

idx im going to sleep.