

- Java model finished training without error
- Vocabulary doesn't contain word "large", probably due to minWordFrequency of 200
- Import error for gensim library
- Ran java model with 300 features vector and 100 minWordFrequency

- Error on model building, did not finish writing model to txt file.
- Running the model builder again
- Reading "A unified model for word sense rep and des" while model is being built
- Cant unzip word2vec example corpus
- Ran python similarity queries under small artificial corpus
- Training python model with 2gb of Medline abstracts

- Java model could not be built, throwing the same error:

```
Exception in thread "main"  
java.lang.IllegalArgumentException: Unable to get linear  
index >= 100  
    at  
org.nd4j.linalg.api.ndarray.BaseNDArray.getDouble(BaseNDArray.java:3229) at  
org.deeplearning4j.models.embeddings.loader.WordVectorSerializer.writeFullModel(WordVectorSerializer.java:722)  
    at  
org.deeplearning4j.examples.nlp.word2vec.MedlineVocabBuilder.main(MedlineVocabBuilder.java:65)
```

- Python model trained under 2Gb of abstracts in 2 hours and 10 mins (lets not forget that the java model was being built at the same time!)
- Managed to do similarity and relationship queries with this model
- Script does not automatically remove punctuation from words
- Programmed Python script to remove punctuation from awk output

- Rebuilt model with punctuation less corpus (40 mins!)
- Results considerably different

- Converted input to lowercase
- Decided to use Snowball Stemmer over Porter and Lancaster, since it seemed better overall if compared to Porter and not as aggressive as Lancaster, which often lead to meaningless words.
- Applied POS tagging to help lemmatizer know when lemmatizing a verb or not
- Finished pre-processor script, running Lemmatizer and Stemmer to medLineAbstract outputs

- Lemmatized and Stemmed output finished without problem
- Left gerunds (handling) and derivational terms (successful)
- Finished downloading Pubmed papers. All papers in directories starting with 'R-Z' had permission denied
- After 4 hours lemmatized and stemmer had not finished (weren't getting processor time), so I restarted the process.

- Finished stemming and lemmatizing Medline corpus. Building models from both files
- Concatenated PubMedSentences into one file ("outputPubMed.txt")
- Stemming and lemmatizing PubMed output
- Some plurals are still present inside Medline lemmatized corpus
- Found way to retrieve and operate on word vectors

- Converted PubMed Corpus to from latin-1 to utf-8
- Pubmed Model built (300 vector size) for both stemmed and lemmatized corpus

-Rebuilding Medline model using 300 vector size

Part of speech get rid of adjectives!?

```
In [11]: model.most_similar(['clock','gene'], ['time'],  
topn=10)
```

```
Out[11]:
```

```
[(u'abdb', 0.5090895891189575),  
 (u'tra2', 0.4717015027999878),  
 (u'meiosisspecific', 0.4659665524959564),  
 (u'mig6', 0.46591469645500183),  
 (u'clockwork', 0.4650305211544037),  
 (u'cry1', 0.46349799633026123),  
 (u'ubx', 0.46010643243789673),  
 (u'rd29a', 0.457570880651474),  
 (u'gnas', 0.4571872055530548),  
 (u'sir2', 0.45700156688690186)]
```

-Downloaded tagged pubMed corpora

-Write the vocabulary size with new tag

-Word2vec in mallet

-Check dpt (vac or protein) example

-Training model on tagged corpus

-Vocab size: collected 2933621 word types from a corpus of 281375791 raw words and 11794233 sentences

-Didn't find relevant results for word2vec in mallet as features

-Initialize sense vectors: for each sense, the candidate

words in the gloss whose average will initialize the sense vector are all **nouns, verbs, adjectives and adverbs** that have a cosine similarity greater than threshold and that are not the word whose sense we are building
-(writing vector to model useful for using similarity queries)
-Large gene/protein/adjective not solved by POS tagging

```
julia> nearest_neighbors(vm, dict, "bank/nn", 1, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("nation/nn",1,0.76438105f0)
("undp/nn",1,0.7573241f0)
("unfpa/nn",1,0.74872416f0)
("unicef/nn",1,0.74308425f0)
("dfid/nn",1,0.73481774f0)
("panamerican/nn",1,0.71827745f0)
("emro/nn",1,0.71280736f0)
("fao/nn",1,0.699815f0)
("ratify/vbd",1,0.6991917f0)
("afro/nn",1,0.69868964f0)
```

```
julia> nearest_neighbors(vm, dict, "bank/nn", 2, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("developmental/jj",5,0.88309246f0)
("hybridoma/nn",1,0.8538016f0)
("iowa/nn",1,0.82617563f0)
```

```
("dshb/nn",1,0.7631244f0)
("city/nn",3,0.67334527f0)
("birmingham/nn",1,0.66670287f0)
("antiÎ²galactosidase/nn",1,0.6485828f0)
("mf20/nn",1,0.6455009f0)
("lexington/nn",1,0.6352473f0)
("bellinzona/nn",1,0.6241812f0)
```

```
julia> nearest_neighbors(vm, dict, "bank/nn", 3, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("alds/nn",1,0.6475176f0)
 ("acroqol/nn",1,0.61817944f0)
 ("raqol/nn",1,0.61817324f0)
 ("embraced/jj",1,0.61793053f0)
 ("campaigner/nns",1,0.6166531f0)
 ("rhetoric/nn",1,0.6138253f0)
 ("lawmaker/nns",1,0.603454f0)
 ("nles/nn",1,0.6015824f0)
 ("reluctantly/rb",1,0.59889024f0)
 ("thrust/vbn",1,0.5977128f0)
```

```
julia> nearest_neighbors(vm, dict, "bank/nn", 4, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("core/nn",3,0.64677453f0)
("tochigi/nn",1,0.59186417f0)
("doÃ±ana/nn",1,0.58450943f0)
("manitoba/nn",1,0.5809157f0)
("ancona/nn",1,0.5757097f0)
("mdch/nn",1,0.57466567f0)
("kumba/nn",1,0.56981784f0)
("extirpate/vbn",1,0.5678243f0)
("nicd/nn",2,0.5642103f0)
("bank/nns",1,0.5603062f0)
```

```
julia> nearest_neighbors(vm, dict, "bank/nn", 5, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("genebank/nn",1,0.7172021f0)
 ("image/nn",5,0.6981371f0)
 ("flj/nn",1,0.6820531f0)
 ("sequenceverified/jj",1,0.66916347f0)
 ("rafl/nn",1,0.66815346f0)
 ("riken/nn",1,0.6643114f0)
 ("identifier/nn",1,0.65953124f0)
 ("transposontagged/jj",1,0.6586819f0)
 ("sorter/nn",2,0.65444183f0)
 ("humanpsd/nn",1,0.6538947f0)
```

Tag organizations in corpus and make them one_word

Stanford – NER
'_' between NNP

Check the clusterings that genia doesn't understand
Cluster based on separate tags

do on a small text sample

replace number with some *num code

check amount of each tag
(date tag doesn't handles all examples)

use genia tags and stanford labels to cluster words and
apply encodings

crawl over wikipedia – alias and gloss for word2vec senses

–NER on pubmed output
–Need to find strategy to join words and then train on the
new corpus

Look for NNPs and see if IN in the end is followed by
capitalized Noun
highlight special cases

–NER tagging didn't finish again, more than a day running
and only 22MB of tagged text...

Traceback (most recent call last):

File "stanfordNER.py", line 26, in <module>

```

    for line in infile:
        File "/anaconda/lib/python2.7/codecs.py", line 699, in
        next
            return self.reader.next()

        File "/anaconda/lib/python2.7/codecs.py", line 630, in
        next
            line = self.readline()

        File "/anaconda/lib/python2.7/codecs.py", line 545, in
        readline
            data = self.read(readsize, firstline=True)

        File "/anaconda/lib/python2.7/codecs.py", line 492, in
        read
            newchars, decodedbytes = self.decode(data, self.errors)

UnicodeDecodeError: 'utf8' codec can't decode bytes in
position 0-1: invalid continuation byte

```

Make tagger and NER file have same number of lines
 Start with only clustering NNP and NNPs connected to
 Capitalized NN (is it what NER does?)
 Check if some of the NN should be clustered with nearby NNP
 Check if tagger don't work, try with chunks

```

julia> disambiguate(vm, dict, "clock", split(""))

```

```

10-element Array{Float64,1}:

```


0.123075
0.467984
0.356351
0.0525896
0.0
0.0
0.0
0.0
0.0
0.0

```
julia> 10-element Array{Float64,1}:
```

ERROR: syntax: extra token "Array" after end of expression

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens were sent for frozenpathology according to the  
wise method"))
```

ERROR: KeyError: were not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens"))
```

```
10-element Array{Float64,1}:
```

0.0423053

0.694691

0.256562

0.00644152

0.0

0.0

0.0

0.0

0.0

0.0

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens were sent for frozen pathology according to the  
wise method"))
```

ERROR: KeyError: were not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens were sent for frozen pathology according to the  
method"))
```

ERROR: KeyError: were not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens"))
```

10-element Array{Float64,1}:

0.0423053

0.694691

0.256562

0.00644152

0.0

0.0

0.0

0.0

0.0

0.0

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens were sent for "))
```

ERROR: KeyError: were not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens be sent for "))
```

ERROR: KeyError: be not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("pathologic
```

```
specimens be send for "))
```

```
ERROR: KeyError: be not found
```

```
in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/  
util.jl:258 (repeats 2 times)
```

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimen be send for "))
```

```
ERROR: KeyError: be not found
```

```
in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/  
util.jl:258 (repeats 2 times)
```

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimen"))
```

```
10-element Array{Float64,1}:
```

```
0.0443191
```

```
0.436385
```

```
0.420631
```

```
0.0986644
```

```
0.0
```

```
0.0
```

```
0.0
```

```
0.0
```

```
0.0
```

```
0.0
```

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens were"))
```

ERROR: KeyError: were not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens sent for frozen pathology according to the  
method"))
```

ERROR: KeyError: for not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens sent"))
```

10-element Array{Float64,1}:

0.0129955

0.510343

0.20222

0.274442

0.0

0.0

0.0

0.0

0.0

0.0

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens sent for"))
```

ERROR: KeyError: for not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens sent frozen pathology according method"))
```

ERROR: KeyError: according not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens sent frozen"))
```

10-element Array{Float64,1}:

0.00336976

0.208896

0.139385

0.648349

0.0

0.0

0.0

0.0

0.0

0.0

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens sent frozen pathology"))
```

10-element Array{Float64,1}:

0.000254355

0.146771

0.427055

0.42592

0.0

0.0

0.0

0.0

0.0

0.0

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens sent frozen pathology according"))
```

ERROR: KeyError: according not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens sent frozen pathology method"))
```

10-element Array{Float64,1}:

0.000883473

0.0232116

0.48516

0.490745

0.0

0.0

0.0

0.0

0.0

0.0

```
julia> disambiguate(vm, dict, "clock", split("pathologic  
specimens sent frozen pathology method"))
```

10-element Array{Float64,1}:

0.000883473

0.0232116

0.48516

0.490745

0.0

0.0

0.0

0.0

0.0

0.0


```
julia> disambiguate(vm, dict, "clock", split(""))
```

```
10-element Array{Float64,1}:
```

```
0.123075
```

```
0.467984
```

```
0.356351
```

```
0.0525896
```

```
0.0
```

```
0.0
```

```
0.0
```

```
0.0
```

```
0.0
```

```
0.0
```

```
julia> nearest_neighbors(vm, dict, "clock", 1, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("evolution",7,0.6554473f0)
```

```
("relaxed",1,0.6331473f0)
```

```
("molecular",2,0.6238983f0)
```

```
("dating",1,0.6134585f0)
```

```
("humanape",1,0.6060164f0)
```

```
("relaxedclock",1,0.5942799f0)
```

```
("phylogenetics",1,0.59347576f0)
```

```
("calibrate",3,0.57518876f0)
("splitting",3,0.5726321f0)
("evolutionary",4,0.558899f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 2, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("circadian",1,0.85195446f0)
 ("rhythm",1,0.7766624f0)
 ("bmal1",1,0.76688135f0)
 ("oscillator",1,0.75553644f0)
 ("rhythmic",1,0.7504024f0)
 ("pacemaker",2,0.72139037f0)
 ("clock",3,0.72121024f0)
 ("clockwork",1,0.71260774f0)
 ("scn",1,0.70764107f0)
 ("oscillation",1,0.6989164f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 3, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("oscillator",1,0.7937242f0)
 ("rhythm",1,0.78263813f0)
 ("circadian",2,0.777422f0)
 ("timekeeping",1,0.76845896f0)
```

```
("circadian",1,0.7519866f0)
("clock",2,0.72121024f0)
("pacemaker",2,0.71248215f0)
("entrainment",1,0.7053692f0)
("rhythmic",1,0.70147526f0)
("rhythmicity",1,0.6740843f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 4, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("idle",1,0.51627976f0)
 ("wagon",1,0.51511955f0)
 ("synchronize",3,0.50358534f0)
 ("desk",1,0.5010427f0)
 ("beep",1,0.49925837f0)
 ("night",4,0.49876022f0)
 ("timer",1,0.49202472f0)
 ("extinguish",2,0.4907013f0)
 ("trolley",1,0.4850198f0)
 ("computer",5,0.4811215f0)
```

```
julia> disambiguate(vm, dict, "clock", split("examined
expression components tumors node negativebreast cancer
patients untreated neoadjuvant adjuvant settings"))
```

```
ERROR: KeyError: tumors not found
```

```
in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
```

util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("examined  
expression components node negativebreast cancer patients  
untreated neoadjuvant adjuvant settings"))
```

ERROR: KeyError: negativebreast not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("examined  
expression components node negative breast cancer patients  
untreated neoadjuvant adjuvant settings"))
```

ERROR: KeyError: patients not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("examined  
expression components node negative breast cancer patient  
untreated neoadjuvant adjuvant settings"))
```

10-element Array{Float64,1}:

0.00016859

0.0424867

0.0925956

0.864749

0.0

0.0

0.0

0.0

0.0

0.0

```
julia> nearest_neighbors(vm, dict, "clock", 4, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("idle",1,0.51627976f0)
 ("wagon",1,0.51511955f0)
 ("synchronize",3,0.50358534f0)
 ("desk",1,0.5010427f0)
 ("beep",1,0.49925837f0)
 ("night",4,0.49876022f0)
 ("timer",1,0.49202472f0)
 ("extinguish",2,0.4907013f0)
 ("trolley",1,0.4850198f0)
 ("computer",5,0.4811215f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 2, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("circadian",1,0.85195446f0)
 ("rhythm",1,0.7766624f0)
 ("bmal1",1,0.76688135f0)
 ("oscillator",1,0.75553644f0)
```

```
("rhythmic",1,0.7504024f0)
("pacemaker",2,0.72139037f0)
("clock",3,0.72121024f0)
("clockwork",1,0.71260774f0)
("scn",1,0.70764107f0)
("oscillation",1,0.6989164f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 3, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("oscillator",1,0.7937242f0)
 ("rhythm",1,0.78263813f0)
 ("circadian",2,0.777422f0)
 ("timekeeping",1,0.76845896f0)
 ("circadian",1,0.7519866f0)
 ("clock",2,0.72121024f0)
 ("pacemaker",2,0.71248215f0)
 ("entrainment",1,0.7053692f0)
 ("rhythmic",1,0.70147526f0)
 ("rhythmicity",1,0.6740843f0)
```

```
julia> disambiguate(vm, dict, "clock", split("Higher
expression several clock genes CLOCK PER1 PER2 PER3 CRY2
NPAS2 RORC found associated longer MFS univariate Cox
regression analyses"))
```

```
ERROR: KeyError: Higher not found
```

```
in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/  
util.jl:258 (repeats 2 times)
```

```
julia> disambiguate(vm, dict, "clock", split("higher  
expression several clock genes CLOCK PER1 PER2 PER3 CRY2  
NPAS2 RORC found associated longer MFS univariate Cox  
regression analyses"))
```

ERROR: KeyError: genes not found

```
in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/  
util.jl:258 (repeats 2 times)
```

```
julia> disambiguate(vm, dict, "clock", split("higher  
expression several clock gene CLOCK PER1 PER2 PER3 CRY2  
NPAS2 RORC found associated longer MFS univariate Cox  
regression analyses"))
```

ERROR: KeyError: CLOCK not found

```
in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/  
util.jl:258 (repeats 2 times)
```

```
julia> disambiguate(vm, dict, "clock", split("higher  
expression several clock gene clock perl per2 per3 cry2  
npas2 rorc found associated longer mfs univariate Cox  
regression analyses"))
```

ERROR: KeyError: Cox not found

```
in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/  
util.jl:258 (repeats 2 times)
```

```
julia> disambiguate(vm, dict, "clock", split("higher  
expression several clock gene clock perl per2 per3 cry2  
npas2 rorc found associated longer mfs univariate cox  
regression analyses"))
```

ERROR: KeyError: analyses not found

in disambiguate at /Users/rberna2/.julia/v0.4/AdaGram/src/
util.jl:258 (repeats 2 times)

```
julia> disambiguate(vm, dict, "clock", split("higher  
expression several clock gene clock per1 per2 per3 cry2  
npas2 rorc found associated longer mfs univariate cox  
regression analyse"))
```

10-element Array{Float64,1}:

8.44201e-13

0.999952

4.76875e-5

3.41268e-17

0.0

0.0

0.0

0.0

0.0

0.0

Stanford create extra tokens for + at end of word (CP45+)
Line breaking differently
Ignore extra lines

take numbers as tokens and remove
remove lsb, rsb, lrb, rrb
write concatenated words to file

was:

the directory_of_open_access_journal lrb doaj rrb list 1358
journal with in excess of 61129 article these number are
increasing on a daily basis

became

the directory_of_open_access_journal doaj list journal with
in excess of article these number are increasing on a daily
basis

*Remove concatenation from entities between punctuation
(commas)
-Check for Birkhaug Moscow*

Build Model with the concatenated corpus

Run POS tagger on CytoVa to allow using it on POS tagged
julia model

Build a script to desambiguate gene names from CytoVa
sentences using all 3 models (concatenated, POS tagged and
normal)

- Split corpus by gene name
- Break line at each sentence
- Run word sense
- Remove punctuations and words from stop words list
- Desambiguate each line

**vm, dict = load_model("/Users/rberna2/.julia/v0.4/AdaGram/
pubMedFinalModel");**

julia> expected_pi(vm, dict.word2id["clock"])

5-element Array{Float64,1}:

0.195534

0.698865

0.0423723

0.0632162

1.26922e-5

julia> nearest_neighbors(vm, dict, "clock", 1, 10)

10-element Array{Tuple{AbstractString,Int64,Float32},1}:

("coalescent",1,0.7314429f0)

("phylogenetics",1,0.7237005f0)

("neutral",5,0.6793695f0)

("relaxed",1,0.66391194f0)

("evolution",1,0.64775026f0)

("phylogeny",2,0.6384808f0)

("humanape",1,0.6358538f0)

("estimation",1,0.6212062f0)

("model",1,0.61612654f0)

("dating",1,0.6153f0)

julia> nearest_neighbors(vm, dict, "clock", 2, 10)

10-element Array{Tuple{AbstractString,Int64,Float32},1}:

("circadian",2,0.8930983f0)

```
("rhythm",1,0.8410981f0)
("oscillator",1,0.8391124f0)
("clockwork",1,0.8092583f0)
("pacemaker",1,0.77402556f0)
("timekeeping",1,0.7527125f0)
("temperatureentrained",1,0.7438298f0)
("entrainment",1,0.7438051f0)
("rhythmicity",1,0.7331676f0)
("rhythmic",1,0.7283246f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 3, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("idle",1,0.72028947f0)
 ("timed",1,0.692858f0)
 ("track",2,0.6831961f0)
 ("flickering",1,0.68224996f0)
 ("switch",5,0.68031204f0)
 ("beep",1,0.6783833f0)
 ("stationary",4,0.6782018f0)
 ("timeout",1,0.67587847f0)
 ("instant",1,0.6682231f0)
 ("stylus",1,0.66735053f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 4, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("zimp10",1,0.75381243f0)
 ("uastik",1,0.7448246f0)
 ("cry1",1,0.7325766f0)
 ("nhr25",1,0.72909135f0)
 ("rorc",1,0.7268255f0)
 ("pdp1",1,0.72473115f0)
 ("csnk1e",1,0.71842104f0)
 ("unc120",1,0.71305156f0)
 ("fbxl3",1,0.7090183f0)
 ("sharp1",1,0.70724285f0)
```

```
vm, dict = load_model("/Users/rberna2/.julia/v0.4/AdaGram/
modelPubMed");
```

```
julia> expected_pi(vm, dict.word2id["clock"])
5-element Array{Float64,1}:
 0.140652
 0.720447
 0.138888
 1.15865e-5
 1.15708e-6
```

```
julia> nearest_neighbors(vm, dict, "clock", 1, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("evolutionary",2,0.7042594f0)
 ("phylogenetics",1,0.70263463f0)
 ("relaxed",2,0.6914519f0)
```

```
("coalescent",1,0.6773237f0)
("calibration",3,0.6532601f0)
("codonsubstitution",1,0.6502222f0)
("neutral",1,0.6474072f0)
("substitution",5,0.6433448f0)
("evolution",3,0.6415082f0)
("molecular",4,0.63312644f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 2, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("circadian",1,0.86938435f0)
 ("rhythm",2,0.79503965f0)
 ("oscillator",1,0.78867996f0)
 ("clockcontrolled",1,0.75497913f0)
 ("bmal1",1,0.7508958f0)
 ("temperatureentrained",1,0.7491481f0)
 ("clockwork",1,0.7466527f0)
 ("rhythmicity",1,0.7438723f0)
 ("rhythmic",1,0.7429863f0)
 ("pdfsecreting",1,0.7346174f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 3, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("entrainment",1,0.731682f0)
 ("phaseshift",1,0.72839975f0)
 ("timekeeping",1,0.71452093f0)
 ("reset",1,0.70255387f0)
 ("oscillator",1,0.6972761f0)
 ("entrain",1,0.69119203f0)
 ("synchronize",2,0.678981f0)
 ("zeitgebers",1,0.67211723f0)
 ("entrained",1,0.67173016f0)

 ("freerun",1,0.67115617f0)
```

```
vm, dict = load_model("/Users/rberna2/.julia/v0.4/AdaGram/
pubMedFinalModel");
```

```
julia> expected_pi(vm, dict.word2id["clock"])
```

```
5-element Array{Float64,1}:
```

0.195534
0.698865
0.0423723
0.0632162
1.26922e-5

```
julia> nearest_neighbors(vm, dict, "clock", 1, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("coalescent",1,0.7314429f0)
 ("phylogenetics",1,0.7237005f0)
 ("neutral",5,0.6793695f0)
 ("relaxed",1,0.66391194f0)
 ("evolution",1,0.64775026f0)
 ("phylogeny",2,0.6384808f0)
 ("humanape",1,0.6358538f0)
 ("estimation",1,0.6212062f0)
 ("model",1,0.61612654f0)
 ("dating",1,0.6153f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 2, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("circadian",2,0.8930983f0)
 ("rhythm",1,0.8410981f0)
```

```
("oscillator",1,0.8391124f0)
("clockwork",1,0.8092583f0)
("pacemaker",1,0.77402556f0)
("timekeeping",1,0.7527125f0)
("temperatureentrained",1,0.7438298f0)
("entrainment",1,0.7438051f0)
("rhythmicity",1,0.7331676f0)
("rhythmic",1,0.7283246f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 3, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("idle",1,0.72028947f0)
("timed",1,0.692858f0)
("track",2,0.6831961f0)
("flickering",1,0.68224996f0)
("switch",5,0.68031204f0)
("beep",1,0.6783833f0)
("stationary",4,0.6782018f0)
("timeout",1,0.67587847f0)
("instant",1,0.6682231f0)
("stylus",1,0.66735053f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 4, 10)
```

10-element Array{Tuple{AbstractString,Int64,Float32},1}:

("zimp10",1,0.75381243f0)

("uastik",1,0.7448246f0)

("cry1",1,0.7325766f0)

("nhr25",1,0.72909135f0)

("rorc",1,0.7268255f0)

("pdp1",1,0.72473115f0)

("csnk1e",1,0.71842104f0)

("unc120",1,0.71305156f0)

("fbxl3",1,0.7090183f0)

("sharp1",1,0.70724285f0)

-AdaGram possibly removes underscored words

-Changed underscore to slash

-Try to project the n dimensions into 2 for visualizing clusters

-Changed tokenize.sh

24Â°_c

-Read directories through Julia

```
julia> vm, dict = load_model("/Users/rberna2/.julia/v0.4/  
AdaGram/pubMedFinalModel2");
```

```
julia> nearest_neighbors(vm, dict, "open_access", 1, 10)
```

ERROR: KeyError: open_access not found


```
in nearest_neighbors at /Users/rberna2/.julia/v0.4/  
AdaGram/src/util.jl:228
```

```
julia> nearest_neighbors(vm, dict, "open access", 1, 10)
```

```
ERROR: KeyError: open access not found
```

```
in nearest_neighbors at /Users/rberna2/.julia/v0.4/  
AdaGram/src/util.jl:228
```

```
julia> nearest_neighbors(vm, dict, "open", 1, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("miniopen",1,0.63408834f0)
```

```
("vertebroplasty",1,0.6298655f0)
```

```
("onlay",1,0.58540154f0)
```

```
("underwent",3,0.5813483f0)
```

```
("pyeloplasty",1,0.5802457f0)
```

```
("laparoscopically",1,0.57955694f0)
```

```
("duraplasty",1,0.57730556f0)
```

```
("prospective",3,0.577229f0)
```

```
("onestage",2,0.5752772f0)
```

```
("laparoscopic",1,0.5743126f0)
```

```
julia> nearest_neighbors(vm, dict, "openaccess", 1, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("publishing",1,0.78181666f0)
```

```
("journal/of/neuroinflammation",1,0.77913624f0)
("subscriptionbased",1,0.7722056f0)
("publisher",1,0.76402736f0)
("open/access",1,0.7619827f0)
("plo/biology",1,0.7534663f0)
("public/library/of/science",1,0.7444576f0)
("biomed/central",1,0.7412007f0)
("plo/clinical/trial",1,0.7300886f0)
("selfarchiving",1,0.7094381f0)
```

```
julia> nearest_neighbors(vm, dict, "open/access", 1, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("journal/of/neuroinflammation",1,0.7880089f0)
 ("biomed/central",1,0.7625581f0)
 ("openaccess",1,0.7619827f0)
 ("publishing",1,0.7603986f0)
 ("publisher",1,0.7499245f0)
 ("public/library/of/science",1,0.7245214f0)
 ("subscriptionbased",1,0.70940036f0)
 ("journal/of/translational/medicine",1,0.70068824f0)
 ("subscription",1,0.7000481f0)
 ("hrj",1,0.68041146f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 1, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("humanape",1,0.65843785f0)
 ("molecular",2,0.60428923f0)
 ("phylogenetics",1,0.5970726f0)
 ("evolution",3,0.5902426f0)
 ("phylogeny",2,0.55106956f0)
 ("relaxed",2,0.5492696f0)
 ("relaxedclock",1,0.5314182f0)
 ("phylogeny",3,0.5293456f0)
 ("divergence",3,0.5188553f0)
 ("evolutionary",1,0.5134894f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 2, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("circadian",1,0.88194877f0)
 ("oscillator",1,0.77712065f0)
 ("scn",1,0.7698663f0)
 ("pacemaker",2,0.7575715f0)
 ("bmal1",1,0.7575045f0)
 ("clock",4,0.7432128f0)
 ("rhythm",2,0.74296415f0)
 ("cry",3,0.7249716f0)
```

```
("rhythmic",1,0.7103827f0)
("rhythmicity",1,0.7062854f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 3, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("lighted",1,0.5671231f0)
 ("photocell",1,0.564114f0)
 ("timer",1,0.5619313f0)
 ("beep",1,0.55937904f0)
 ("screwed",1,0.5576059f0)
 ("skate",2,0.55596256f0)
 ("placing",4,0.5545787f0)
 ("waited",2,0.5540759f0)
 ("descended",2,0.5486632f0)
 ("tripod",1,0.54545677f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 4, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("circadian",3,0.7786537f0)
 ("oscillator",1,0.772293f0)
 ("rhythm",2,0.7606397f0)
 ("circadian",1,0.75320256f0)
 ("clock",2,0.7432127f0)
```

```
("timekeeping",1,0.7431088f0)
("pacemaker",2,0.7336834f0)
("entrainment",1,0.6645678f0)
("rhythmicity",1,0.65790606f0)
("clockwork",1,0.6565481f0)
```

```
julia> nearest_neighbors(vm, dict, "clock", 5, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("node",5,-Inf32)
 ("node",5,-Inf32)
 ("node",5,-Inf32)
 ("node",5,-Inf32)
 ("node",5,-Inf32)
 ("node",5,-Inf32)
 ("node",5,-Inf32)
 ("node",5,-Inf32)
 ("node",5,-Inf32)
 ("node",5,-Inf32)
```

```
julia> expected_pi(vm, dict.word2id["clock"])
5-element Array{Float64,1}:
 0.179511
 0.497636
```

0.0288619

0.293978

1.29408e-5

```
julia> expected_pi(vm, dict.word2id["bad"])
```

5-element Array{Float64,1}:

0.0678676

0.107755

0.316769

0.26934

0.238268

```
julia> nearest_neighbors(vm, dict, "clock", 5, 10)
```

^[[D10-element Array{Tuple{AbstractString,Int64,Float32},1}:

("node",5,-Inf32)

("node",5,-Inf32)

("node",5,-Inf32)

("node",5,-Inf32)

("node",5,-Inf32)

("node",5,-Inf32)

("node",5,-Inf32)

("node",5,-Inf32)

("node",5,-Inf32)

```
("node",5,-Inf32)
```

```
julia> nearest_neighbors(vm, dict, "bad", 1, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("excellent",2,0.7438963f0)
```

```
("good",2,0.7166653f0)
```

```
("fair",2,0.67802f0)
```

```
("poor",2,0.6466123f0)
```

```
("rated",2,0.56748533f0)
```

```
("dichotomised",1,0.5661116f0)
```

```
("poor0",1,0.52529675f0)
```

```
("scale",1,0.5214183f0)
```

```
("poor1",1,0.52128637f0)
```

```
("scored",5,0.5183529f0)
```

```
julia> nearest_neighbors(vm, dict, "bad", 2, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("flagged",1,0.5711415f0)
```

```
("flagging",1,0.5710342f0)
```

```
("unreliable",1,0.522904f0)
```

```
("rejected",4,0.52259624f0)
```

```
("spot",1,0.49617556f0)
```

```
("flag",3,0.49505582f0)
```

```
("filtering",1,0.48861742f0)
("manually",1,0.48041937f0)
("badly",1,0.47968143f0)
("procat",1,0.47876003f0)
```

```
julia> nearest_neighbors(vm, dict, "bad", 3, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("proapoptotic",1,0.7845363f0)
 ("bak",1,0.7722325f0)
 ("bcl2",2,0.76517797f0)
 ("bclxl",1,0.757724f0)
 ("bax",1,0.7440859f0)
 ("antiapoptotic",1,0.73988557f0)
 ("bid",2,0.7393184f0)
 ("bclw",1,0.71859694f0)
 ("bh3only",1,0.71538603f0)
 ("bim",1,0.70690835f0)
```

```
julia> nearest_neighbors(vm, dict, "bad", 4, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("something",1,0.7303726f0)
 ("thing",1,0.72271645f0)
 ("stuff",1,0.7087126f0)
```



```
("nothing",1,0.7069315f0)
("want",1,0.70656466f0)
("put",3,0.6969281f0)
("anything",1,0.695243f0)
("really",3,0.692184f0)
("somebody",1,0.6895754f0)
("saying",1,0.6840634f0)
```

```
julia> nearest_neighbors(vm, dict, "bad", 5, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("luck",1,0.57739455f0)
 ("selffulfilling",1,0.5694204f0)
 ("arrogance",1,0.565762f0)
 ("good",5,0.5568181f0)
 ("trump",1,0.55652034f0)
 ("materialist",1,0.5493076f0)
 ("worse",4,0.5486719f0)
 ("slippery",2,0.5465239f0)
 ("condemn",1,0.54415077f0)
 ("virtuous",1,0.5405478f0)
```

```
julia> nearest_neighbors(vm, dict, "dpt", 1, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("tttt",1,0.45582983f0)
("rmd",1,0.4557201f0)
("yk",2,0.4434464f0)
("mkv2",1,0.44091117f0)
("sechellia",1,0.42320034f0)
("sg1",1,0.42267084f0)
("purinepyrimidine",1,0.4198303f0)
("ttt",2,0.41950104f0)
("mlh",1,0.40668973f0)
("a86",1,0.40344116f0)
```

```
julia> nearest_neighbors(vm, dict, "dpt", 2, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("menc",2,0.727537f0)
("measlesmumpsrubella",1,0.7215535f0)
("dtp",1,0.7076965f0)
("dtap",1,0.702096f0)
("dos",3,0.69728976f0)
("dpt3",1,0.6829246f0)
("opv",1,0.6723775f0)
("hib",1,0.66042686f0)
("polio",1,0.6353888f0)
("seroresponse",1,0.62826484f0)
```



```
("node",5,-Inf32)
```

```
("node",5,-Inf32)
```

```
("node",5,-Inf32)
```

```
julia> expected_pi(vm, dict.word2id["dpt"])
```

```
5-element Array{Float64,1}:
```

```
0.336145
```

```
0.662427
```

```
0.001319
```

```
9.93906e-5
```

```
9.93899e-6
```

```
julia> expected_pi(vm, dict.word2id["large"])
```

```
5-element Array{Float64,1}:
```

```
0.213481
```

```
0.160993
```

```
0.147231
```

```
0.199123
```

```
0.279172
```

```
julia> nearest_neighbors(vm, dict, "large", 1, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("small",1,0.80076134f0)
```

```
("numerous",5,0.66122574f0)
("larger",3,0.6395037f0)
("often",1,0.6124756f0)
("formed",5,0.6059311f0)
("cell",2,0.5909802f0)
("tiny",1,0.5907699f0)
("mediumsized",1,0.58953035f0)
("dncdk5expressing",1,0.5882197f0)
("occasionally",3,0.58335143f0)
```

```
julia> ^C
```

```
julia> nearest_neighbors(vm, dict, "large", 2, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
("small",5,0.7553482f0)
("limited",1,0.5998586f0)
("many",3,0.57548296f0)
("smaller",1,0.56080043f0)
("larger",4,0.55185336f0)
("numerous",4,0.54698074f0)
("total",4,0.545796f0)
("whole",3,0.5346206f0)
("mrassociated",1,0.5312529f0)
```

```
("frnas",1,0.52462584f0)
```

```
julia> nearest_neighbors(vm, dict, "large", 3, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("small",3,0.5974622f0)
```

```
("containing",4,0.5453512f0)
```

```
("chloroplastunique",1,0.49850026f0)
```

```
("including",3,0.4970664f0)
```

```
("include",5,0.49000067f0)
```

```
("eg",5,0.48966637f0)
```

```
("gtpase/ran",1,0.48808742f0)
```

```
("particular",1,0.4876473f0)
```

```
("limited",3,0.48483562f0)
```

```
("also",3,0.48155448f0)
```

```
julia> nearest_neighbors(vm, dict, "large", 4, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("small",2,0.81230795f0)
```

```
("relatively",2,0.67410475f0)
```

```
("sufficiently",2,0.6288281f0)
```

```
("larger",1,0.62528867f0)
```

```
("therefore",3,0.58059496f0)
```

```
("rather",1,0.5759698f0)
```

```
("fact",3,0.57515574f0)
("since",4,0.57294506f0)
("hence",4,0.56737f0)
("quite",1,0.564042f0)
```

```
julia> nearest_neighbors(vm, dict, "large", 5, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("small",4,0.7958927f0)
 ("larger",2,0.70042217f0)
 ("smaller",4,0.6344747f0)
 ("limited",2,0.5723764f0)
 ("megafauna",1,0.56256515f0)
 ("relatively",5,0.547253f0)
 ("orangutan",2,0.5307232f0)
 ("eg",2,0.5239313f0)
 ("kinabatangan",1,0.52109647f0)
 ("largest",4,0.52067715f0)
```

```
julia> nearest_neighbors(vm, dict, "small", 2, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("large",4,0.81230795f0)
 ("relatively",2,0.65837085f0)
 ("larger",1,0.6554369f0)
```

```
("thus",3,0.6553966f0)
("therefore",3,0.6531542f0)
("fact",3,0.6476153f0)
("even",5,0.6276825f0)
("size",2,0.6231274f0)
("spikecount",1,0.6191683f0)
("expected",2,0.6177407f0)
```

```
julia> nearest_neighbors(vm, dict, "small", 3, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("rna",3,0.6384254f0)
 ("large",3,0.59746224f0)
 ("interfering",1,0.59532386f0)
 ("piwiinteracting",1,0.578993f0)
 ("molecule",1,0.5729548f0)
 ("interference",1,0.56737715f0)
 ("mirnp",1,0.55975056f0)
 ("noncoding",3,0.55330336f0)
 ("silencing",1,0.5505572f0)
 ("rasirna",1,0.53601766f0)
```

```
julia> nearest_neighbors(vm, dict, "large", 3, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
```


("small",3,0.5974622f0)
("containing",4,0.5453512f0)
("chloroplastunique",1,0.49850026f0)
("including",3,0.4970664f0)
("include",5,0.49000067f0)
("eg",5,0.48966637f0)
("gtpase/ran",1,0.48808742f0)
("particular",1,0.4876473f0)
("limited",3,0.48483562f0)
("also",3,0.48155448f0)
("syntaxinbinding",1,0.47975928f0)
("dhcs",2,0.474876f0)
("multidomain",2,0.47289395f0)
("known",4,0.4703014f0)
("largest",2,0.47003567f0)
("48s",1,0.46851748f0)
("adenoviral",3,0.4681374f0)
("ypl184c",1,0.46662247f0)
("substraterecognition",1,0.4646828f0)
("example",5,0.4638754f0)
("comprise",4,0.46370882f0)
("multisubunit",1,0.46344343f0)
("initialize",2,0.46337315f0)

```
("composed",1,0.46337143f0)
("assembles",1,0.46333534f0)
("thought",2,0.46180424f0)
("nsp10",1,0.46163622f0)
("nsp9",1,0.46069676f0)
("likely",1,0.46053413f0)
("associated",5,0.46001092f0)
```

```
julia> nearest_neighbors(vm, dict, "containing", 4, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("contains",1,0.74083567f0)
("contain",3,0.67586887f0)
("composed",1,0.6447858f0)
("termed",3,0.63204277f0)
("domain",5,0.62775654f0)
("consists",4,0.6115755f0)
("repeat",3,0.61114156f0)
("like",1,0.6056751f0)
("comprising",1,0.5849202f0)
("ranbinding",1,0.5801918f0)
("glutaminerich",1,0.5674352f0)
("called",1,0.5649141f0)
("consist",3,0.56300855f0)
```

```
("ddw",2,0.56102115f0)
("carboxylterminal",1,0.5609232f0)
("math",3,0.5602585f0)
("consisting",3,0.55937517f0)
("cysteinerich",1,0.5592132f0)
("singlepass",2,0.5573126f0)
("fyve",1,0.55706525f0)
("cterminal",2,0.5553834f0)
("motif",2,0.5545704f0)
("homologous",2,0.55323327f0)
("four",3,0.55281633f0)
("wd40",1,0.54802305f0)
("roc2",1,0.5476459f0)
("rich",1,0.5470175f0)
("carboxyterminal",1,0.5467926f0)
("aminoterminal",1,0.5466333f0)
("large",3,0.545351f0)
```

```
julia> vm[dict.word2id["dpt"]]
```

```
ERROR: MethodError: `getindex` has no method matching  
getindex(::AdaGram.VectorModel, ::Int32)
```

```
julia> vm.code[dict.word2id["dpt"]]
```

```
-1
```

```
julia> vm.out[dict.word2id["dpt"]]
```

```
ERROR: type VectorModel has no field out
```

```
julia> vm.Out[dict.word2id["dpt"]]
```

```
0.00043494615f0
```

```
julia> dict.word2id["dpt"]
```

```
26557
```

```
julia> dict[dict.word2id["dpt"]]
```

```
ERROR: MethodError: `getindex` has no method matching  
getindex(::AdaGram.Dictionary, ::Int32)
```

```
julia> vm.frequencies[dict.word2id["dpt"]]
```

```
212
```

```
julia> vm.code[dict.word2id["dpt"]]
```

```
-1
```

```
julia> vm.path[dict.word2id["dpt"]]
```

```
0
```

```
julia> vm.d[dict.word2id["dpt"]]
```

ERROR: BoundsError

in getindex at number.jl:15

julia> vm[dict.word2id["dpt"]]

**ERROR: MethodError: `getindex` has no method matching
getindex(::AdaGram.VectorModel, ::Int32)**

julia> vm[dict.word2id["dpt"]]

**ERROR: MethodError: `getindex` has no method matching
getindex(::AdaGram.VectorModel, ::Int32)**

julia> nearest_neighbors(vm, dict, "bad_homburg", 1, 30)

ERROR: KeyError: bad_homburg not found

**in nearest_neighbors at /Users/rberna2/.julia/v0.4/
AdaGram/src/util.jl:228**

julia> nearest_neighbors(vm, dict, "bad/homburg", 1, 30)

30-element Array{Tuple{AbstractString,Int64,Float32},1}:

("fresenius/kabi",1,0.7463762f0)

("fresenius",1,0.7284239f0)

("pulsion/medical/system/ag",1,0.6860505f0)

("ingelheim",1,0.6859997f0)

("melsungen",1,0.6857363f0)

("b/braun",1,0.6597047f0)

("ludwigshafen",1,0.6542841f0)

("neuss",1,0.65253925f0)
("munich",1,0.64688003f0)
("gmbh",1,0.6304559f0)
("schering",1,0.61695474f0)
("dreieich",1,0.6165094f0)
("polysulfone",1,0.61374295f0)
("reutlingen",1,0.6116088f0)
("bayer/ag",1,0.603776f0)
("mbrecht",1,0.60109776f0)
("leverkusen",1,0.5957026f0)
("lake/forest",1,0.5954903f0)
("basf",1,0.5954077f0)
("healthcare",5,0.59419954f0)
("bracco",1,0.5900919f0)
("parkedavis",1,0.5892811f0)
("mexico",4,0.5855936f0)
("mallinckrodt",1,0.5840152f0)
("beerse",1,0.5819346f0)
("norderstedt",1,0.58011824f0)
("merck/kgaa",1,0.579339f0)
("sanofi",1,0.57918936f0)
("janssen/pharmaceutica",1,0.5777064f0)
("teflonâ",1,0.57437736f0)

```
julia> nearest_neighbors(vm, dict, "bad/homburg", 2, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("cysteine",2,0.23972532f0)
 ("serine",3,0.23941295f0)
 ("endopeptidases",1,0.23715602f0)
 ("peptidase",2,0.22620505f0)
 ("fi",4,0.22471021f0)
 ("protease",5,0.22202958f0)
 ("nutrition",3,0.21493119f0)
 ("proteinase",2,0.21009137f0)
 ("game",3,0.20834477f0)
 ("dis",2,0.20822908f0)
 ("electroneutrality",1,0.20813936f0)
 ("accordance",1,0.20806995f0)
 ("randomly",1,0.2076917f0)
 ("transmembrane",2,0.20499213f0)
 ("acr",1,0.20429315f0)
 ("secretory",2,0.20391616f0)
 ("permittivity",1,0.20235229f0)
 ("governs",1,0.20185055f0)
 ("metalloproteases",1,0.19852237f0)
 ("protease",2,0.19821656f0)
```

```
("case12",1,0.19746903f0)
("subtilisinlike",1,0.1969567f0)
("parenteral",1,0.19580416f0)
("european/prospective/investigation",1,0.19535851f0)
("zstack",1,0.19519953f0)
("fp1",2,0.1947947f0)
("birtoxinlike",1,0.1938229f0)
("proteinsâ",1,0.19328228f0)
("gmax",1,0.19296111f0)
("american/rheumatism/association",1,0.19245975f0)
```

-train concatenated model with grammar tag
-don't forget 300 features, stop words and both 5 and 10 senses

-previous model verbs were not lemmatized properly (POS tag is capitalized and comparison was not capitalized)

```
julia> vm, dict = load_model("/Users/rberna2/.julia/v0.4/AdaGram/pubMedTaggedFinalModel300");
```

```
julia> expected_pi(vm, dict.word2id["clock"])
```

ERROR: KeyError: clock not found

in getindex at dict.jl:724


```
julia> expected_pi(vm, dict.word2id["clock/nn"])
```

```
5-element Array{Float64,1}:
```

```
0.164674
```

```
0.766691
```

```
0.0462756
```

```
0.0223443
```

```
1.48634e-5
```

```
julia> expected_pi(vm, dict.word2id["clock/nns"])
```

```
5-element Array{Float64,1}:
```

```
0.941304
```

```
0.0585939
```

```
9.28153e-5
```

```
8.43762e-6
```

```
8.43806e-7
```

```
julia> nearest_neighbors(vm, dict, "clock/nns", 1, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("rhythm/nn",4,0.767521f0)
```

```
("circadian/nn",1,0.74298716f0)
```

```
("clock/nn",2,0.73823535f0)
```

```
("timekeeping/nn",1,0.72542715f0)
```

```
("circadian/jj",2,0.7166037f0)
("oscillator/nn",1,0.7130727f0)
("oscillator/nns",1,0.71068186f0)
("circadian/jj",1,0.68143857f0)
("pacemaker/nn",2,0.67081285f0)
("rhythmic/jj",1,0.6444746f0)
```

```
julia> nearest_neighbors(vm, dict, "clock/nns", 2, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("locoh/nn",1,0.6373494f0)
 ("j/vbp",3,0.60936224f0)
 ("qdb/nnp",1,0.60597336f0)
 ("keffective/nnp",1,0.6034727f0)
 ("weighed/vbn",3,0.59704626f0)
 ("mlc/nnp",2,0.5964199f0)
 ("fì/nn",1,0.5929934f0)
 ("kì/nn",1,0.58775544f0)
 ("coalhmm/nn",1,0.5875424f0)
 ("csmet/nn",1,0.5858561f0)
```

```
julia> nearest_neighbors(vm, dict, "clock/nn", 1, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("humanape/nn",1,0.6390415f0)
```

```
("relaxed/nn",1,0.62034994f0)
("relaxed/jj",1,0.5977605f0)
("uf/nnp",3,0.5968904f0)
("phylogenetics/nns",1,0.5773721f0)
("clocklike/nn",1,0.57596475f0)
("phylogeny/nn",1,0.57458186f0)
("dating/nn",1,0.5685828f0)
("phylogeny/nns",1,0.5634872f0)
("molecular/jj",1,0.5505895f0)
```

```
julia> nearest_neighbors(vm, dict, "clock/nn", 2, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("circadian/jj",1,0.8705302f0)
 ("rhythm/nn",4,0.84652776f0)
 ("oscillator/nn",1,0.8201027f0)
 ("rhythmic/jj",1,0.7793377f0)
 ("circadian/nn",1,0.7729185f0)
 ("pacemaker/nn",2,0.75824267f0)
 ("rhythmicity/nn",1,0.7524804f0)
 ("circadian/jj",2,0.7451292f0)
 ("clock/nns",1,0.7382352f0)
 ("entrainment/nn",1,0.7335508f0)
```

```
julia> nearest_neighbors(vm, dict, "clock/nn", 3, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("turned/vbn",4,0.5992223f0)
 ("timer/nn",1,0.58046407f0)
 ("30/cd",3,0.5671649f0)
 ("reset/vbn",1,0.5563744f0)
 ("day/nn",2,0.5552647f0)
 ("sunset/vb",1,0.5522051f0)
 ("beginning/nn",1,0.54057837f0)
 ("start/nn",4,0.53980225f0)
 ("run/nns",5,0.5376909f0)
 ("min/jj",4,0.53485113f0)
```

```
julia> nearest_neighbors(vm, dict, "clock/nn", 4, 10)
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("rethink/vb",1,0.6308228f0)
 ("workâ/nn",2,0.62614846f0)
 ("facile/jj",2,0.62605095f0)
 ("unfolding/vbg",2,0.6257232f0)
 ("mannerâ/nn",1,0.6237956f0)
 ("overstretched/vbn",1,0.6226473f0)
 ("contemplate/vbp",1,0.6210652f0)
 ("teleological/jj",1,0.61454475f0)
```

```
("interrupt/vbz",2,0.6139847f0)
```

```
("brighter/nn",2,0.6123318f0)
```

```
julia> expected_pi(vm, dict.word2id["bad/nns"])
```

```
ERROR: KeyError: bad/nns not found
```

```
in getindex at dict.jl:724
```

```
julia> expected_pi(vm, dict.word2id["bad/nn"])
```

```
ERROR: KeyError: bad/nn not found
```

```
in getindex at dict.jl:724
```

```
julia> expected_pi(vm, dict.word2id["bad/jj"])
```

```
5-element Array{Float64,1}:
```

```
0.081252
```

```
0.243634
```

```
0.292083
```

```
0.0952394
```

```
0.287791
```

```
julia> expected_pi(vm, dict.word2id["bad/jjs"])
```

```
ERROR: KeyError: bad/jjs not found
```

```
in getindex at dict.jl:724
```

```
julia> expected_pi(vm, dict.word2id["bad/jj"])
```

```
5-element Array{Float64,1}:
```

```
0.081252
```

```
0.243634
```

```
0.292083
```

```
0.0952394
```

```
0.287791
```

```
julia> expected_pi(vm, dict.word2id["bad/nn"])
```

```
ERROR: KeyError: bad/nn not found
```

```
in getindex at dict.jl:724
```

```
julia> nearest_neighbors(vm, dict, "bad/jj", 1, 10)
```

```
10-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("flagged/vbd",1,0.67220753f0)
```

```
("flagged/vbn",1,0.6653273f0)
```

```
("spot/nns",1,0.6271589f0)
```

```
("flagged/jj",1,0.61942124f0)
```

```
("snr/jj",1,0.5528122f0)
```

```
("flagging/vbg",1,0.54423136f0)
```

```
("unreliably/rb",1,0.5411519f0)
```

```
("backgroundsubtracted/jj",1,0.5275233f0)
```

```
("signaltobackground/nn",1,0.5247819f0)
```

```
("background/vb",1,0.51498544f0)
```

```
julia> nearest_neighbors(vm, dict, "clock/nns", 2, 120)
```

```
^C
```

```
ERROR: InterruptException:
```

```
  in getindex at /Applications/Julia-0.4.5.app/Contents/  
Resources/julia/lib/julia/sys.dylib
```

```
  in nearest_neighbors at /Users/rberna2/.julia/v0.4/  
AdaGram/src/util.jl:222
```

```
  in nearest_neighbors at /Users/rberna2/.julia/v0.4/  
AdaGram/src/util.jl:229
```

```
julia>
```

```
julia> nearest_neighbors(vm, dict, "clock/nns", 2, 30)
```

```
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
 ("locoh/nn",1,0.6373494f0)
```

```
 ("j/vbp",3,0.60936224f0)
```

```
 ("qdb/nnp",1,0.60597336f0)
```

```
 ("keffective/nnp",1,0.6034727f0)
```

```
 ("weighed/vbn",3,0.59704626f0)
```

```
 ("mlc/nnp",2,0.5964199f0)
```

```
 ("fì/nn",1,0.5929934f0)
```

```
 ("kì/nn",1,0.58775544f0)
```

```
 ("coalhmm/nn",1,0.5875424f0)
```

("csmet/nn",1,0.5858561f0)
("alignmentâ/nn",1,0.5854725f0)
("converge/vbp",2,0.58300674f0)
("alocoh/nn",1,0.5783237f0)
("appendix_s3/nnp",1,0.5772871f0)
("sequel/nn",2,0.5762717f0)
("esst/nn",1,0.5756249f0)
("episcan/nnp",1,0.5749604f0)
("coalescentbased/vbn",1,0.5702452f0)
("csmet/nnp",1,0.5694676f0)
("noisy_add/nnp",1,0.5684368f0)
("gr/nn",5,0.56815827f0)
("hapmix/nnp",1,0.5659786f0)
("prescribed/vbn",4,0.56511945f0)
("valuefunctions/nns",1,0.56459653f0)
("tan/nnp",2,0.5645067f0)
("margarita/nnp",1,0.5617515f0)
("eqtn/nn",1,0.5607332f0)
("mstmap/nnp",1,0.55884475f0)
("marginalizing/vbg",1,0.55871737f0)
("employ/vbp",3,0.5583423f0)

julia> expected_pi(vm, dict.word2id["clock/nnp"])

5-element Array{Float64,1}:

0.999332

0.000607238

5.51931e-5

5.01787e-6

5.01822e-7

julia> nearest_neighbors(vm, dict, "clock/nnp", 1, 30)

30-element Array{Tuple{AbstractString,Int64,Float32},1}:

("bmal1/nn",1,0.7731021f0)

("bmal1/jj",1,0.76633275f0)

("npas2/jj",1,0.7595874f0)

("per2/nn",1,0.7432729f0)

("per1/nn",1,0.7420657f0)

("npas2/nn",1,0.7305167f0)

("clock/nn",2,0.7172883f0)

("bmal1/vbp",1,0.7037165f0)

("per2/jj",1,0.7016283f0)

("cry2/jj",1,0.68154347f0)

("circadian/jj",1,0.6804321f0)

("rorc/nnp",1,0.67563444f0)

("shortperiod/jj",1,0.67252725f0)

("cry1/nn",1,0.6707666f0)

```
("mper1/nn",1,0.6645508f0)
("cry2/nn",1,0.6617178f0)
("arntl/nnp",1,0.66156787f0)
("mper1/jj",1,0.6615508f0)
("per3/nn",1,0.6594602f0)
("rhythmic/jj",1,0.65887475f0)
("bma1lâ/nn",1,0.65808135f0)
("clk/nn",1,0.65344733f0)
("cry1/jj",1,0.65176755f0)
("clockcontrolled/jj",1,0.63825184f0)
("cry/nn",3,0.6372319f0)
("mper2/jj",1,0.63489246f0)
("clock/vbp",1,0.6297765f0)
("circadian/nn",1,0.6276283f0)
("cryptochrome/nnp",1,0.6258483f0)
("tomato/nn",4,0.6249417f0)
```

```
julia> nearest_neighbors(vm, dict, "clock/nnp", 2, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
```

[illegible]

`("available/jj",5,-Inf32)`

`("available/jj",5,-Inf32)`

`("available/jj",5,-Inf32)`

`julia> nearest_neighbors(vm, dict, "clock/vbp", 1, 30)`

`30-element Array{Tuple{AbstractString,Int64,Float32},1}:`

`("bmal1/vbp",1,0.7350299f0)`

`("clockcontrolled/jj",1,0.68703926f0)`

`("npas2/nn",1,0.67177224f0)`

`("reverbs/nn",1,0.66302407f0)`

`("lightindependent/jj",1,0.65977085f0)`

`("lightregulated/jj",1,0.6543393f0)`

`("rors/nnp",1,0.6532512f0)`

`("tim/vb",1,0.65295255f0)`

`("npas2/jj",1,0.65140957f0)`

`("cry2/jj",1,0.6443844f0)`

`("cryptochromes/nnp",1,0.641361f0)`

`("mper2/jj",1,0.63882643f0)`

`("pertim/nnp",1,0.6337113f0)`

`("pertim/nn",1,0.63071716f0)`

`("rores/nnp",1,0.6300174f0)`

`("clock/nnp",1,0.6297765f0)`

`("noncircadian/jj",1,0.6277821f0)`

("pseudoresponse/nn",1,0.627459f0)
("clock/vb",1,0.6268673f0)
("clk/jj",1,0.62590814f0)
("bma11/vb",1,0.6205736f0)
("anticipates/vbz",2,0.6194619f0)
("decl/nn",1,0.6153429f0)
("clkcyc/nnp",1,0.61005217f0)
("clk/nn",1,0.60573304f0)
("tim/jj",1,0.6049637f0)
("e4bp4/nn",1,0.60493404f0)
("bma11/jj",1,0.6003822f0)
("timekeeping/nn",1,0.5968134f0)
("lhy/jj",1,0.59481084f0)

```
julia> nearest_neighbors(vm, dict, "clock/vbp", 2, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
("periodicity/nn",3,0.26333997f0)
("3year/cd",2,0.24706207f0)
("exam/nn",1,0.24246573f0)
("spread/jj",2,0.24145657f0)
("marchmay/nn",1,0.23894773f0)
("yearly/jj",1,0.23660396f0)
("past/jj",2,0.23484755f0)
```

("whatever/wdt",1,0.23235601f0)
("peaking/vbg",2,0.23036836f0)
("forty/jj",1,0.23013169f0)
("februarymarch/nn",1,0.2299405f0)
("campaign/nn",2,0.22971132f0)
("pfapi/nn",1,0.22745706f0)
("dong_district/nnp",1,0.2263873f0)
("nhaps/nnp",1,0.22572662f0)
("peak/vbz",2,0.22492924f0)
("bimodal/vbn",1,0.22135358f0)
("juneaugust/nn",1,0.22113645f0)
("yearly/rb",1,0.22073235f0)
("circa/nn",1,0.22031084f0)
("inhabitant/nn",1,0.22003986f0)
("cholera/vb",1,0.21941833f0)
("almora/nnp",1,0.21929649f0)
("certification/nn",2,0.21873409f0)
("boukoki/nnp",1,0.21855903f0)
("envenomings/nns",1,0.21844052f0)
("examination/nn",3,0.21778807f0)
("french_west_africa/nnp",1,0.21720976f0)
("vast/jj",3,0.21707137f0)
("approximately/rb",2,0.21658708f0)

[illegible]

```
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
```

```
julia> expected_pi(vm, dict.word2id["bad/nnp"])
```

```
5-element Array{Float64,1}:
```

```
0.886911
```

```
0.112294
```

```
0.000722615
```

```
6.56851e-5
```

```
6.56851e-6
```

```
julia> expected_pi(vm, dict.word2id["bad/nn"])
```

```
ERROR: KeyError: bad/nn not found
```

```
in getindex at dict.jl:724
```



```
julia> expected_pi(vm, dict.word2id["bad/nns"])
```

```
ERROR: KeyError: bad/nns not found
```

```
in getindex at dict.jl:724
```

```
julia> expected_pi(vm, dict.word2id["bad/jj"])
```

```
5-element Array{Float64,1}:
```

```
0.081252
```

```
0.243634
```

```
0.292083
```

```
0.0952394
```

```
0.287791
```

```
julia> nearest_neighbors(vm, dict, "bad/nnp", 1, 30)
```

```
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("bad/jj",3,0.7614155f0)
```

```
("bclw/nnp",1,0.754661f0)
```

```
("bid/nn",2,0.7500204f0)
```

```
("bclxl/jj",1,0.7387883f0)
```

```
("nox/nnp",1,0.73727405f0)
```

```
("bclxl/vb",1,0.73622507f0)
```

```
("bak/nnp",1,0.723922f0)
```

```
("bclxl/nnp",1,0.7237039f0)
```

```
("puma/nnp",1,0.720443f0)
```

```
("bim/nnp",2,0.71827227f0)
("bclw/nn",1,0.7119068f0)
("proapoptotic/jj",1,0.7062446f0)
("subgroup/jj",3,0.69673634f0)
("bclxl/nn",1,0.6939989f0)
("bh3only/rb",1,0.6867343f0)
("bh3only/jj",1,0.6859722f0)
("bik/nnp",1,0.6831411f0)
("mcl1/nn",1,0.6822393f0)
("bcl2/nns",1,0.6669998f0)
("antiapoptotic/jj",1,0.6650349f0)
("bak/nn",1,0.65787977f0)
("ciap1/jj",1,0.6568126f0)
("bclx/nn",1,0.6551758f0)
("bcl2/vbd",1,0.655151f0)
("bcl2/nn",1,0.65453935f0)
("bax/nnp",1,0.65293586f0)
("diablo/nnp",1,0.64980334f0)
("bimel/nnp",1,0.64899856f0)
("bh3/nn",1,0.64267766f0)
("ciap1/nn",1,0.64110345f0)
```

```
julia> nearest_neighbors(vm, dict, "bad/nnp", 2, 30)
```

30-element Array{Tuple{AbstractString,Int64,Float32},1}:

("singleitem/jj",1,0.63271827f0)
("identified/vb",2,0.6280818f0)
("firstperson/jj",1,0.62581974f0)
("kqol26/nn",1,0.6206836f0)
("undertaken/in",1,0.6186809f0)
("asthmaspecific/jj",1,0.61863494f0)
("satisfaction/jj",1,0.61835873f0)
("pdq39/nn",1,0.61693066f0)
("pgwbi/nnp",1,0.6161438f0)
("parentproxy/jj",1,0.613528f0)
("addqol/nnp",1,0.6133441f0)
("checklist/jj",1,0.61293477f0)
("37item/cd",1,0.6126902f0)
("itqol/nnp",1,0.6122793f0)
("rlsqol/nnp",1,0.61167395f0)
("macdqol/nn",1,0.61111253f0)
("ppme/nnp",1,0.61107653f0)
("stomaqol/nn",1,0.6086006f0)
("surveybased/jj",1,0.6074885f0)
("hui/nn",1,0.60748184f0)
("mncas/nnp",1,0.6072131f0)
("abim/nnp",1,0.60648394f0)


```
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
```

```
julia> nearest_neighbors(vm, dict, "bad/jj", 1, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("flagged/vbd",1,0.67220753f0)
 ("flagged/vbn",1,0.6653273f0)
```

("spot/nns",1,0.6271589f0)
("flagged/jj",1,0.61942124f0)
("snr/jj",1,0.5528122f0)
("flagging/vbg",1,0.54423136f0)
("unreliably/rb",1,0.5411519f0)
("backgroundsubtracted/jj",1,0.5275233f0)
("signaltobackground/nn",1,0.5247819f0)
("background/vb",1,0.51498544f0)
("intensity/vbz",1,0.50673914f0)
("rejected/vbn",4,0.5054143f0)
("backgroundcorrected/jj",1,0.50223446f0)
("manually/rb",3,0.50087357f0)
("backgroundsubtracted/vbn",1,0.4986367f0)
("unreliable/jj",1,0.4984684f0)
("flagging/nn",1,0.49821776f0)
("spot/nn",2,0.49300587f0)
("spot/jj",1,0.49199367f0)
("spot/vbp",1,0.48938644f0)
("flag/nns",1,0.48857737f0)
("gridding/nn",1,0.48810402f0)
("artifact/nns",2,0.48672298f0)
("maplots/nns",1,0.48513207f0)
("rejection/nn",4,0.48410445f0)

```
("background/vbn",1,0.483595f0)
("thresholded/jj",1,0.48351598f0)
("bluefuse/nnp",1,0.48294675f0)
("printtip/jj",1,0.47907424f0)
("excluded/vbn",1,0.4765757f0)
```

```
julia> nearest_neighbors(vm, dict, "bad/jj", 2, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
("horrible/jj",1,0.6974998f0)
("evil/jj",1,0.68907875f0)
("nt/dt",1,0.6874927f0)
("scared/vbn",1,0.68085855f0)
("scared/vbd",1,0.68031853f0)
("scared/jj",1,0.6746843f0)
("think/vbp",2,0.6740161f0)
("terrible/jj",1,0.67380726f0)
("going/vbg",2,0.67047334f0)
("something/nn",1,0.6699931f0)
("thing/nn",1,0.66835684f0)
("say/vb",2,0.6673119f0)
("awful/jj",1,0.66627246f0)
("everything/nn",1,0.6662005f0)
("s/vbz",5,0.6613825f0)
```

```
("oh/in",2,0.6593162f0)
("maybe/rb",2,0.6588869f0)
("feel/vbp",1,0.6584238f0)
("thing/nns",1,0.6571429f0)
("get/vbp",4,0.6569113f0)
("sort/nn",2,0.6562262f0)
("afraid/nn",1,0.6560352f0)
("okay/nn",1,0.65551984f0)
("yeah/nn",1,0.65339583f0)
("anything/nn",1,0.6527808f0)
("scary/jj",1,0.65228206f0)
("nasty/jj",1,0.65192956f0)
("nt/rb",1,0.651222f0)
("stuff/nn",1,0.65015304f0)
("m/md",1,0.6485391f0)
```

```
julia> nearest_neighbors(vm, dict, "bad/jj", 3, 30)
```

```
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("proapoptotic/jj",1,0.80107814f0)
("bid/nn",2,0.7894515f0)
("bak/nnp",1,0.76651037f0)
("bcl2/nn",1,0.7624903f0)
("bad/nnp",1,0.76141566f0)
```


("antiapoptotic/jj",1,0.7560041f0)
("bclxl/nn",1,0.74530745f0)
("bim/nnp",2,0.7411092f0)
("bcl2/jj",1,0.73920137f0)
("bclw/nnp",1,0.7287502f0)
("bax/nnp",1,0.7219203f0)
("nox/nnp",1,0.7174212f0)
("bcl2/nns",1,0.7170776f0)
("subgroup/jj",3,0.71576744f0)
("bclxl/nnp",1,0.71377796f0)
("bh3only/rb",1,0.70652544f0)
("mcl1/nn",1,0.70439243f0)
("puma/nnp",1,0.6950819f0)
("bclxl/jj",1,0.69175404f0)
("bclxl/vb",1,0.67362016f0)
("bclw/nn",1,0.6716315f0)
("bclx/nn",1,0.667661f0)
("bcl2/vb",1,0.6669662f0)
("bax/nn",1,0.660634f0)
("bh3only/jj",1,0.65881747f0)
("xiap/nnp",1,0.6412538f0)
("bcl2/nnp",1,0.6406203f0)
("bcl2/vbp",1,0.6399838f0)

("bik/nnp",1,0.6397085f0)

("bh3/nn",1,0.63044596f0)

julia> nearest_neighbors(vm, dict, "bad/jj", 5, 30)

30-element Array{Tuple{AbstractString,Int64,Float32},1}:

("good/jj",2,0.6066796f0)

("patently/rb",1,0.6038484f0)

("reputation/nns",1,0.6014846f0)

("truthâ/nn",1,0.59326667f0)

("foolish/jj",1,0.5927463f0)

("inherited/vbn",4,0.5926613f0)

("factsâ/nn",1,0.59225434f0)

("believer/nns",1,0.5897487f0)

("sanctity/nn",1,0.584359f0)

("themâ/nn",1,0.58432233f0)

("excuse/nn",1,0.58283997f0)

("denier/nns",1,0.58256143f0)

("vanity/nn",1,0.5816139f0)

("admitted/vbn",3,0.5808403f0)

("cabot/nnp",1,0.57689106f0)

("unfriendly/jj",1,0.57534516f0)

("â/cc",2,0.5750117f0)

("cardano/nnp",1,0.57430506f0)

```
("believer/nn",1,0.57180566f0)
("arrogance/nn",1,0.571634f0)
("â/rb",4,0.57097995f0)
("disrespect/nn",1,0.5707106f0)
("powerless/jj",1,0.56807876f0)
("beauty/nn",2,0.56804466f0)
("miracle/nns",1,0.56771547f0)
("disservice/nn",1,0.5671417f0)
("wielgus/nnp",1,0.5665763f0)
("courtroom/nn",1,0.566287f0)
("idea/nn",5,0.56548715f0)
("itâ/vb",1,0.56527936f0)
```

```
julia> nearest_neighbors(vm, dict, "bad/jj", 4, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("good/jj",1,0.78447056f0)
 ("poor/jj",5,0.7367316f0)
 ("excellent/jj",2,0.7360467f0)
 ("fair/nn",1,0.7314795f0)
 ("excellent/nn",1,0.7160364f0)
 ("fair/jj",1,0.6894057f0)
 ("dichotomised/vbn",1,0.60297614f0)
 ("excellent/vb",1,0.5967347f0)
```

("dichotomized/vbn",1,0.5782755f0)
("good/nnp",1,0.57362396f0)
("poor0/nn",1,0.5639997f0)
("selfrated/vbn",1,0.5540145f0)
("rated/vbd",1,0.5540105f0)
("rated/vbn",2,0.54812473f0)
("fair/jj",4,0.54361564f0)
("fair/vb",1,0.5267258f0)
("5point/jj",1,0.5048286f0)
("likert/nn",1,0.48831463f0)
("dissatisfied/vbd",1,0.47911057f0)
("6point/cd",1,0.4783363f0)
("rated/vbn",1,0.4781302f0)
("5point/cd",1,0.47355005f0)
("dichotomous/jj",2,0.47305077f0)
("selfrated/vbd",1,0.47240853f0)
("likert/jj",1,0.47226772f0)
("good/nn",1,0.4715023f0)
("worst/jjs",2,0.4675194f0)
("satisfied/jj",1,0.46744144f0)
("fivepoint/jj",1,0.46692714f0)
("scale/nn",1,0.46565083f0)

```
julia> expected_pi(vm, dict.word2id["large/jj"])
```

```
5-element Array{Float64,1}:
```

```
0.166288
```

```
0.28199
```

```
0.217303
```

```
0.132471
```

```
0.201948
```

```
julia> expected_pi(vm, dict.word2id["large/nn"])
```

```
ERROR: KeyError: large/nn not found
```

```
in getindex at dict.jl:724
```

```
julia> expected_pi(vm, dict.word2id["large/nns"])
```

```
ERROR: KeyError: large/nns not found
```

```
in getindex at dict.jl:724
```

```
julia> expected_pi(vm, dict.word2id["large/nnp"])
```

```
5-element Array{Float64,1}:
```

```
0.997829
```

```
0.00197367
```

```
0.000179409
```

```
1.63099e-5
```

```
1.63099e-6
```

```
julia> nearest_neighbors(vm, dict, "large/nnp", 1, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("saleh/nnp",1,0.81126696f0)
 ("humeau/nnp",1,0.6898356f0)
 ("1993a/cd",1,0.6812017f0)
 ("tamarro/nnp",1,0.67767155f0)
 ("2008b/cd",1,0.67713404f0)
 ("albert/nnp",1,0.6759688f0)
 ("xiao/nnp",2,0.66509587f0)
 ("hatanaka/nnp",1,0.6620976f0)
 ("inoue/nnp",1,0.6588168f0)
 ("hirashima/nnp",1,0.65754706f0)
 ("giguere/nnp",1,0.65723926f0)
 ("ouyang/nnp",1,0.65722275f0)
 ("1995a/cd",1,0.6517891f0)
 ("asakura/nnp",1,0.6496241f0)
 ("gao/nn",1,0.649239f0)
 ("frazier/nnp",1,0.6448648f0)
 ("burzomato/nnp",1,0.64197487f0)
 ("1997b/cd",1,0.641952f0)
 ("laganiere/nnp",1,0.64161277f0)
 ("kallen/nnp",1,0.6393516f0)
```

```
("noguchi/nnp",1,0.6359014f0)
("chong/nnp",1,0.63424647f0)
("riehle/nnp",1,0.6338661f0)
("obata/nnp",1,0.63217765f0)
("jansson/nnp",1,0.6315129f0)
("1993b/cd",1,0.63045603f0)
("beutner/nnp",1,0.628728f0)
("ikegaya/nnp",1,0.6284381f0)
("liu/nnp",2,0.62776977f0)
("shin/nnp",1,0.6277282f0)
```

```
julia> nearest_neighbors(vm, dict, "large/nnp", 2, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
("voltageactivated/jj",1,0.23640774f0)
("voltagegated/vbd",1,0.23494792f0)
("bkca/nnp",1,0.23481695f0)
("sugita/nnp",1,0.2303786f0)
("cited/vbn",3,0.22672826f0)
("trpm8/nn",1,0.22423305f0)
("bcl2/nn",2,0.22370894f0)
("coexpressing/vbg",1,0.22156978f0)
("sac/nnp",1,0.22054325f0)
("colocalize/vbp",2,0.22025824f0)
```

```
("breast_cancer_research/nnp",1,0.21942881f0)
("trpc7/nn",1,0.21929336f0)
("voltagegated/nn",1,0.218651f0)
("bcl2/vbp",1,0.21776897f0)
("voltagegated/jj",1,0.2171767f0)
("hclca1/nn",1,0.21714741f0)
("voltagegated/vbn",1,0.21684892f0)
("cd19/nn",1,0.21646363f0)
("reporting/vbg",2,0.21554676f0)
("cruz/nnp",1,0.21552338f0)
("kv/nn",2,0.21533327f0)
("cd68/nn",1,0.21517687f0)
("perforin/nn",1,0.21287455f0)
("hdhko/nn",1,0.21261671f0)
("paper/nns",2,0.21094784f0)
("dn/nn",3,0.20905864f0)
("receptorassociated/vbd",2,0.20905776f0)
("explicitly/rb",3,0.20888148f0)
("electroacupuncture/nn",2,0.20868893f0)
("nonselective/jj",1,0.20828591f0)
```

```
julia> nearest_neighbors(vm, dict, "large/nnp", 3, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
```


[illegible]

```
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
```

```
julia> nearest_neighbors(vm, dict, "large/jj", 1, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
("small/jj",5,0.78653866f0)
("relatively/rb",1,0.6018211f0)
("larger/jjr",1,0.5766437f0)
("huref/nn",1,0.5647048f0)
("unigene/nnp",3,0.54812753f0)
("numerous/jj",3,0.5431954f0)
("smaller/jjr",5,0.5426554f0)
("many/jj",5,0.54229456f0)
("hglbs/nnp",1,0.5360925f0)
("limited/jj",1,0.5323735f0)
("whole/jj",1,0.52531374f0)
("wholegene/nn",1,0.51840293f0)
("re/nnp",3,0.51352125f0)
```

("reduces/nns",5,0.5115973f0)
("represent/vbp",4,0.50269604f0)
("ufw/nnp",1,0.50234383f0)
("frnas/nns",1,0.50017136f0)
("copy/nn",5,0.49975613f0)
("signifying/vbg",3,0.4987612f0)
("rrmi/nnp",1,0.49837753f0)
("chimpanzeespecific/jj",1,0.4937934f0)
("svg/nnp",2,0.49261332f0)
("ccgh/nn",1,0.49145523f0)
("therefore/rb",3,0.4910124f0)
("short/jj",5,0.4908968f0)
("pairedends/nns",1,0.48957485f0)
("limited/vbn",1,0.4891289f0)
("microinversions/nns",1,0.4891165f0)
("ecuniculi/nnp",1,0.48878777f0)
("ignored/vbd",2,0.48820105f0)

```
julia> nearest_neighbors(vm, dict, "large/jj", 2, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("larger/jjr",2,0.7185748f0)
 ("smaller/jjr",3,0.64546365f0)
 ("relatively/rb",2,0.61021066f0)
```

("small/jj",3,0.5906728f0)
("limited/vbn",5,0.5677779f0)
("megafauna/nn",1,0.56455326f0)
("limited/jj",3,0.5597239f0)
("thus/rb",4,0.55389357f0)
("btb/nn",3,0.55384046f0)
("kinabatangan/nnp",1,0.54059756f0)
("example/nn",1,0.5311035f0)
("furthermore/rb",3,0.5305408f0)
("bird/nn",2,0.5294053f0)
("largest/jjs",5,0.5212654f0)
("wild/jj",5,0.51733834f0)
("also/rb",1,0.515526f0)
("shortnose/jj",1,0.5146877f0)
("rare/jj",5,0.51459354f0)
("many/jj",2,0.5101219f0)
("caribbean/nnp",2,0.5098125f0)
("bird/nns",4,0.50672394f0)
("seabird/nns",1,0.5049793f0)
("small/jj",4,0.50262326f0)
("therefore/rb",1,0.50180715f0)
("caatinga/nnp",1,0.50017977f0)
("hph5n1/nn",1,0.49907142f0)

```
("contrast/nn",4,0.49775252f0)
("condor/nns",1,0.49668738f0)
("size/nns",1,0.49588078f0)
("suggests/vbz",1,0.49533257f0)
```

```
julia> nearest_neighbors(vm, dict, "large/jj", 3, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("small/jj",1,0.8257886f0)
("larger/jjr",3,0.68413746f0)
("smaller/jjr",1,0.6662191f0)
("numerous/jj",5,0.65990525f0)
("oval/vb",1,0.6393268f0)
("vesicle/nns",2,0.62749016f0)
("often/rb",5,0.6190275f0)
("mediumsized/vbn",1,0.6132458f0)
("cluster/nns",1,0.6046208f0)
("electrondense/jj",1,0.60420066f0)
("formed/vbn",4,0.6037891f0)
("formed/vbd",3,0.59135526f0)
("surrounded/vbn",1,0.5895281f0)
("present/jj",5,0.5883209f0)
("cell/nns",2,0.588206f0)
("within/in",1,0.587918f0)
```

```
("prominent/jj",1,0.58602965f0)
("irregular/jj",1,0.5856271f0)
("nest/nns",3,0.5832279f0)
("dense/jj",2,0.57967544f0)
("scattered/vbn",1,0.57841873f0)
("round/vbn",1,0.5782566f0)
("ssecksir/nn",1,0.577712f0)
("ramified/jj",1,0.57672995f0)
("dnck5expressing/vbg",1,0.5763794f0)
("oval/jj",1,0.5760742f0)
("tiny/jj",1,0.57535964f0)
("rounded/vbn",2,0.5749987f0)
("sometimes/rb",4,0.5730673f0)
("scattered/vbd",1,0.57250553f0)
```

```
julia> nearest_neighbors(vm, dict, "large/jj", 4, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("small/jj",2,0.6021406f0)
 ("largest/jjs",1,0.5458068f0)
 ("chloroplastunique/nn",1,0.5446669f0)
 ("encoding/vbg",4,0.51744646f0)
 ("accessory/jj",2,0.5143729f0)
 ("ribosomal/jj",3,0.513231f0)
```

("ribosomeassociated/jj",1,0.5058847f0)
("ublike/nnp",1,0.5055137f0)
("rpp25/nn",1,0.50251025f0)
("deadbox/nnp",1,0.5013998f0)
("chaperone/vbp",1,0.5006384f0)
("diverse/jj",3,0.4994861f0)
("subunit/vbz",2,0.4994818f0)
("comprise/vbp",4,0.4937805f0)
("protein/nns",3,0.49001762f0)
("encode/vbp",1,0.4890691f0)
("eif4a/jj",1,0.4885561f0)
("precludes/in",2,0.48823613f0)
("typically/rb",1,0.48815912f0)
("nlpc/jj",1,0.4863431f0)
("elike/nn",1,0.48397785f0)
("multicomponent/jj",2,0.48387954f0)
("48s/cd",1,0.48383257f0)
("dicerlike/nn",1,0.48355487f0)
("unique/jj",1,0.48224625f0)
("rho/vbd",2,0.48099884f0)
("comprises/vbz",4,0.48062396f0)
("contains/vbz",2,0.47896022f0)
("encoded/vbn",4,0.478782f0)

`("ranbinding/nnp",1,0.4782356f0)`

`julia> nearest_neighbors(vm, dict, "large/jj", 5, 30)`

`30-element Array{Tuple{AbstractString,Int64,Float32},1}:`

`("small/jj",3,0.7284158f0)`

`("smaller/jjr",4,0.630967f0)`

`("larger/jjr",4,0.61098635f0)`

`("therefore/rb",2,0.60391575f0)`

`("spikecount/nn",1,0.58487964f0)`

`("since/in",1,0.56246793f0)`

`("fact/nn",5,0.55946285f0)`

`("negligible/jj",2,0.55293685f0)`

`("sufficiently/rb",5,0.5507554f0)`

`("generally/rb",1,0.54952383f0)`

`("case/nn",3,0.5385424f0)`

`("much/rb",5,0.53799003f0)`

`("expected/vbn",2,0.5358437f0)`

`("insensitive/jj",1,0.5320869f0)`

`("thus/rb",3,0.5311684f0)`

`("reasonable/jj",1,0.5291714f0)`

`("quite/rb",1,0.5275922f0)`

`("becomes/vbz",1,0.52560526f0)`

`("multiinformation/nn",1,0.524458f0)`


```
("ro/nn",4,0.52296156f0)
("size/vbz",1,0.5216406f0)
("converge/vbp",2,0.52116764f0)
("increasing/vbg",2,0.5178223f0)
("sensitive/jj",3,0.51754135f0)
("finite/vb",1,0.51718575f0)
("hence/rb",1,0.516571f0)
("finitesize/jj",1,0.5138528f0)
("possible/jj",1,0.5135434f0)
("kì/nn",1,0.51307786f0)
("selfexcitable/jj",1,0.51201177f0)
```

```
julia> expected_pi(vm, dict.word2id["dpt/nn"])
```

```
5-element Array{Float64,1}:
```

```
0.995669
0.00393705
0.0003579
3.25364e-5
3.25365e-6
```

```
julia> expected_pi(vm, dict.word2id["dpt/nns"])
```

```
ERROR: KeyError: dpt/nns not found
```

```
in getindex at dict.jl:724
```

```
julia> expected_pi(vm, dict.word2id["dpt/nnp"])
```

```
5-element Array{Float64,1}:
```

```
0.485612
```

```
0.498814
```

```
0.0151714
```

```
0.000366228
```

```
3.66221e-5
```

```
julia> expected_pi(vm, dict.word2id["dpt/jj"])
```

```
ERROR: KeyError: dpt/jj not found
```

```
in getindex at dict.jl:724
```

```
julia> nearest_neighbors(vm, dict, "dpt/nn", 1, 30)
```

```
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
```

```
("flat/jj",5,0.55413264f0)
```

```
("mkv2/nn",1,0.54954255f0)
```

```
("herpesvirus/nn",2,0.5366478f0)
```

```
("a0/jj",2,0.5349341f0)
```

```
("manifested/vbn",4,0.5347648f0)
```

```
("pointed/vbn",5,0.5284529f0)
```

```
("anticipated/vbn",5,0.523676f0)
```

```
("please/vb",3,0.51849866f0)
```

("subchain/nn",1,0.5143145f0)
("patchesâ/nn",1,0.509716f0)
("produce/nn",3,0.5079656f0)
("prif/nn",1,0.5075271f0)
("peaked/vbn",3,0.507295f0)
("6fold/cd",3,0.50608873f0)
("younger/jjr",5,0.50535023f0)
("msd/nnp",3,0.5051086f0)
("randomizing/vbg",2,0.50280386f0)
("average/pdt",1,0.5027964f0)
("observes/vbz",2,0.5005819f0)
("reshuffled/vbd",1,0.5004542f0)
("explore/vbp",3,0.50011927f0)
("crossed/vbd",3,0.49584082f0)
("r1/nnp",1,0.4955312f0)
("count/vbz",5,0.49377948f0)
("tenth/jj",2,0.4905328f0)
("landscape/nn",3,0.49051636f0)
("sla/vbz",1,0.49037257f0)
("transfectants/nns",3,0.48720992f0)
("furthermore/nn",4,0.487038f0)
("integrated/vbd",3,0.48571122f0)

```
julia> nearest_neighbors(vm, dict, "dpt/nn", 2, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("branch/nns",4,0.2551631f0)
 ("cochlear/nn",1,0.23878285f0)
 ("platelet/nn",1,0.22194292f0)
 ("platelet/nn",3,0.2207487f0)
 ("aggregation/nn",2,0.2206653f0)
 ("vacv/nn",1,0.21804419f0)
 ("girth/nn",1,0.21653625f0)
 ("collateral/nns",1,0.21352425f0)
 ("clopidogrel/nn",2,0.21174914f0)
 ("identification/vbp",2,0.21163118f0)
 ("beneath/vb",1,0.20938577f0)
 ("tunica/nnp",1,0.2084219f0)
 ("eosinophil/nnp",1,0.20766357f0)
 ("muscleâ/nn",1,0.20609117f0)
 ("thinner/rb",1,0.20551306f0)
 ("acra/nn",1,0.20382342f0)
 ("neurite/jj",2,0.20368955f0)
 ("tubule/jj",1,0.20335436f0)
 ("branching/nn",2,0.20247605f0)
 ("diameter/nn",5,0.20200416f0)
 ("langley/nnp",1,0.20127709f0)
```

("tungsten/nn",1,0.19862595f0)

```
("available/jj",5,-Inf32)
```

```
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
```

```
julia> nearest_neighbors(vm, dict, "dpt/nnp", 1, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
("asbestosexposed/jj",1,0.64451325f0)
```

("plaque/nns",3,0.58844185f0)
("rpf/nnp",2,0.5640562f0)
("micronodular/jj",1,0.520159f0)
("pleural/jj",1,0.51253605f0)
("consolidation/nns",1,0.50941503f0)
("thickening/vbg",1,0.5065508f0)
("asbestosis/nn",1,0.5056993f0)
("emphysema/jj",1,0.50559103f0)
("fibrosis/nn",4,0.50352293f0)
("thickening/nn",1,0.49730536f0)
("extracutaneous/jj",1,0.49476686f0)
("eosinophilic/nnp",1,0.49395582f0)
("herpetiformis/nn",1,0.49094594f0)
("emphysema/nn",1,0.48730716f0)
("noncardiogenic/jj",1,0.4861831f0)
("interstitial/jj",1,0.48428255f0)
("peribronchial/jj",1,0.48366082f0)
("fibrosis/jj",1,0.4807207f0)
("siderosis/nn",1,0.47984356f0)
("duodenitis/nn",1,0.47893512f0)
("submucous/jj",1,0.47878748f0)
("adenopathy/jj",1,0.47872174f0)
("bronchiectasis/nn",1,0.47714162f0)

```
("atopy/jj",1,0.47612035f0)
("pseudotumor/nn",1,0.47561356f0)
("proteinosi/nn",1,0.4747697f0)
("moderate/severe/jj",1,0.4737816f0)
("peliosis/nn",1,0.47218913f0)
("pulmonary/jj",2,0.46782532f0)
```

```
julia> nearest_neighbors(vm, dict, "dpt/nnp", 2, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("measles/mumps/rubella/nn",1,0.81515014f0)
 ("dtp/nn",1,0.8098239f0)
 ("dos/nns",4,0.8065779f0)
 ("menc/nn",2,0.7972409f0)
 ("opv/nnp",1,0.7904447f0)
 ("dtp/nn",1,0.7893373f0)
 ("dtp3/nn",1,0.76486313f0)
 ("hib/nn",1,0.7384165f0)
 ("diphtheria/nn",2,0.73344606f0)
 ("polio/nn",1,0.7300219f0)
 ("vaccine/nn",1,0.72966856f0)
 ("hib/nnp",1,0.7239997f0)
 ("vaccine/vbp",1,0.7144432f0)
 ("tetanus/nn",1,0.69671685f0)
```



```
("vaccination/nn",2,0.6953266f0)
("threedose/jj",1,0.694424f0)
("mmrv/nnp",1,0.6841359f0)
("diphtheria/nns",1,0.6778986f0)
("vaccine/nns",2,0.67450166f0)
("toxoid/nns",1,0.6739221f0)
("measles/vbz",1,0.6694356f0)
("pertussis/nn",2,0.66092086f0)
("pentavalent/jj",1,0.656791f0)
("seroresponse/nn",1,0.6567776f0)
("trivalent/nn",2,0.6531654f0)
("immunization/nns",2,0.6485912f0)
("tetanus/jj",1,0.64438576f0)
("toxoid/nn",1,0.64025146f0)
("twodose/jj",1,0.63702744f0)
("rubella/nn",1,0.63338506f0)
```

```
julia> nearest_neighbors(vm, dict, "dpt/nnp", 3, 30)
30-element Array{Tuple{AbstractString,Int64,Float32},1}:
 ("nlx/nnp",1,0.6565632f0)
 ("mor/nn",2,0.64729285f0)
 ("organotins/nns",1,0.63616264f0)
 ("dtab/nnp",1,0.6273056f0)
```

("f6/jj",2,0.62487334f0)
("orthosteric/jj",1,0.6235045f0)
("advocating/vbg",2,0.6217721f0)
("tae226/nn",1,0.6178617f0)
("hippuristanol/nn",1,0.6152525f0)
("psd95/vb",1,0.6133869f0)
("y101k/nn",1,0.61286026f0)
("camkii/vb",1,0.610831f0)
("prohibit/vb",2,0.60977674f0)
("tbidmt1/nn",1,0.6091069f0)
("underwent/nn",3,0.60796076f0)
("omad2/jj",1,0.60675246f0)
("mpep/nn",1,0.606525f0)
("pp2a/vbp",1,0.6039239f0)
("originated/vbn",4,0.60265344f0)
("8cpt/cd",1,0.6022615f0)
("mad2wt/nn",1,0.6017295f0)
("agonistbound/nn",1,0.6014146f0)
("nshc/jj",1,0.6001693f0)
("feedback/vb",2,0.5986986f0)
("selor/nnp",1,0.597586f0)
("cmad2/vb",1,0.5964768f0)
("distamycin/vbz",1,0.59493816f0)


```
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
("available/jj",5,-Inf32)
```

```
julia> vm.In[:,1,dict.word2id["dpt/jj"]]
ERROR: KeyError: dpt/jj not found
in getindex at dict.jl:724
```

```
julia> vm.In[:,1,dict.word2id["dpt/nnp"]]
300-element Array{Float32,1}:
 0.137557
 0.0502738
 0.0930585
```

-0.211008
-0.0518723
-0.196472
0.0716375
-0.0500634
-0.414992
0.162337
0.0651872
0.222679
0.160393
0.00274799
-0.194063
-0.0859989
0.0953742
0.113451
-0.199936
0.250376
-0.381635
0.10169
-0.243584
:
0.227844
0.168996

-0.205614
0.181389
-0.0356631
0.0264956
0.0950599
-0.226139
-0.0667909
0.0330117
-0.07824
-0.241421
-0.320842
0.18667
-0.0913945
0.0887194
-0.120871
0.166244
-0.0550706
-0.0585599
0.0962138
0.0867226
-0.0474547

Vector corresponding to second prototype of word "apple" can be obtained from `vm.In[:, 2, dict.word2id["apple"]]`

Find subtraction in adagram

```
x = vm.In[:, 1, dict.word2id["apple/nn"]];  
y = vm.In[:, 1, dict.word2id["software/nn"]]
```

```
nearest_neighbors(vm, dict, x/norm(x) + y/norm(y), 10)
```

excluding used words:

```
x = dict.word2id["large/nnp"]  
y = dict.word2id["gene/nn"]
```

```
nearest_neighbors(vm, dict, vm.In[:, 1, x]/norm(vm.In[:, 1, x]) + vm.In[:, 1, y]/norm(vm.In[:, 1, y]), 10, exclude=[(x, 1), (y, 1)])
```

```
julia> x = dict.word2id["mlh/nnp"]
```

```
84259
```

```
julia> y = dict.word2id["gene/nn"]
```

```
8
```

```
julia> z = dict.word2id["kinase/nn"]
```

```
608
```

```
julia> nearest_neighbors(vm, dict, vm.In[:, 1, x]/  
norm(vm.In[:, 1, x]) - vm.In[:, 1, y]/norm(vm.In[:, 1, y])  
+ vm.In[:, 5, z]/norm(vm.In[:, 5, z]), 10, exclude=[(x, 1),
```

(y,1),(z,5)])

10-element Array{Tuple{AbstractString,Int64,Float32},1}:

("cdc2/nn",1,0.8864332f0)

("cdk7/nn",1,0.8687598f0)

("sek/nn",2,0.8616022f0)

("cdks/nnp",1,0.8455092f0)

("cdkactivating/vbg",1,0.84390616f0)

("cdc2/jj",1,0.8415919f0)

("cak/nnp",1,0.83828694f0)

("cdk/nnp",1,0.83645695f0)

("ken/nnp",1,0.8316009f0)

("kip/nnp",1,0.8265572f0)