CoVec

Covington Vector Semantics Tools

Michael A. Covington, Ph.D.

2016 September 19

Covington Innovations

About CoVec

CoVec, the Covington Vector Semantics Tools, is a set of software tools for comparing words and texts using vector semantics. It uses the data files released by the GloVe Project (http://www.nlp.stanford.edu/projects/glove), which are free to use for all purposes.

CoVec will tell you:

- Whether successive words in a text are commonly used together (thus, whether the text is disorganized and frequently changing topic);
- Whether portions of the text of some specified length (such as 15 words long) are likely to contain topic changes;
- Whether all the words in a text are commonly used together (thus, whether the text is about one topic or many);
- How much each of the words in a text resembles each of the others (in this case your text is likely to be just a list of words);
- How much the vocabulary of each of a set of texts resembles the other texts in the set.

The key idea behind vector semantics is that the meaning of a word can be measured by the way it co-occurs with other words. What is measured is not meaning in the traditional sense but is close, to the point that synonyms are easily recognized.

CoVec gets its information about language from GloVe data sets. Slightly simplifying, the way Stanford University's GloVe works is to break an enormous text into short segments using a moving window, and make a table of how many times each word occurs in the same window as each other word. (For example, *dog* and *cat* often occur near each other; *dog* and *geometry,* much less often.) The result is an enormous matrix, with perhaps 100,000 by 100,000 elements or more. The next step is then to use mathematical techniques to reduce the dimensionality of the matrix so that it has only a few hundred columns, so that it is small enough for computation to be feasible.

The similarity of two words is then the vector cosine similarity of their rows in the matrix: 1.0 if they are the same. If the matrix contained only counts, the lowest possible similarity would be 0.0, but due to the dimensionality-reduction algorithm, GloVe matrices can have rows with similarities somewhat below zero.

The largest data set presently released by GloVe, a 5.5-gigabyte file named glove.840B.300d.txt, works well with CoVec. It was produced by analyzing 840 billion words of English text and reducing the dimensionality to 300 columns.

Installation

To run CoVec under Windows, all you need is the file **`CoVec.exe`** and one or more GloVe vector sets in **`.txt`** format.

Optionally, you can use the CoVec installer. It places **`CoVec.exe`** in your system's program files and enables you to launch it from any command prompt, and to uninstall CoVec from Control Panel. You will still need to obtain a GloVe vector set from Stanford.

CoVec is compiled for Windows Vista and later using .NET Client Framework 4. Under MacOS and Linux, CoVec runs as a Mono command-line application.

Basic usage of CoVec

CoVec is a Windows command-line application whose usage is summarized as follows:

```
CoVec -vec vecfile –ascii -in infile -wordseq       -verbose  >outfile  2>logfile
                                     -coherence  n
                                     -wordmatrix
                                     -textmatrix

-vec vecfile  - File of vectors, in GloVe text format.

-ascii        - Optional. Read the vector set in a faster way that does
                not preserve non-ASCII characters.

-in infile    - Text file to be processed; ASCII or UTF-8.
                If a wildcard pattern is used, such as C:\xxx\*.txt,
                all files that match the pattern will be read and
                processed in succession.

Choose one of the following analyses:
-wordseq      - Report similarity of consecutive words in each text.
-coherence n  - Report average similarity to each other of all words in every
                n-word segment of the text to each other, using all positions of
                a moving window.  If n is 0, use the entire text as one window.
-wordmatrix   - Output a matrix of the similarity of each word in the
                text to all the others. (-in must be a wildcard.)
-textmatrix   - Output a matrix of the similarity of each text file to
                all the others. (-in must be a wildcard.)  The similarity
                of a text to itself is not 1; it is that text's coherence.

-verbose      - Optional. Include more information in the output.
```

```
-pause        - Optional. Pause for user to press a key upon finishing.

> outfile     - Optional. Redirect output to the specified file, from
                which Excel or R can read it as space-delimited text.
                Use >> to append to a file that already exists.
                If not redirected, output is written to the screen.

2> logfile    - Optional. Redirect progress messages to the specified file.
                Use 2>> to append to a file that already exists.
                Use 2>&1 to send progress messages to the main output file.
                If not redirected, progress messages are written to the
                screen.
```

For example, the following command performs a word-sequence analysis of all the files whose name matches **`t*.txt`** in the current directory, using **`GloVe.840B.300d.txt`** as the vector set:

```
CoVec -vec GloVe.840B.300d.txt -in t*.txt -wordseq
```

Of course, the file names can include full paths to other directories.

Usage in MacOS and Linux

To use CoVec in MacOS or Linux, install Mono, which is available free of charge from http://www.mono-project.com or through the Linux package system. Under Linux, install the package *mono-complete*; it is common for Linux systems to include part of Mono but not all that is needed.

Then run CoVec from the command line with mono CoVec.exe in place of CoVec, and putting quotes around any file name that contains *. For example, the command example above would look like this (with the differences highlighted):

```
mono CoVec.exe -vec GloVe.840B.300d.txt -in "t*.txt" -wordseq
```

Paths are permitted on all file names, including CoVec.exe.

Input files

The input files for CoVec are plain text (editable in Notepad, not Microsoft Word files). They can be in ASCII, Unicode, or Windows-1252 (ANSI) format.

The language is expected to be English, with normal spelling and punctuation. Misspelled words are not recognized.

CAUTION: If you use the wildcard pattern **`*.txt`** to specify the input files, note that this also matches GloVe vector files, which are enormous and cannot be processed as input. CoVec will skip them because they are too large and are not pure text, but you will see warning messages.

Division of words

CoVec breaks the text into words, splitting contractions (e.g., *we'll* => *we 'll*) and removing punctuation marks other than apostrophes. If you are unsure how particular words are being broken, run CoVec on a relevant sample of text with the **`-wordseq`** and **`-verbose`** options.

Stop words and missing words

CoVec ignores a set of "stop words" that do not indicate subject matter. In the current version, the stop words are:

> *a, able, about, across, after, all, almost, also, am, among,*
> *an, and, any, are, as, at, be, because, been, but, by, can,*
> *cannot, could, dear, did, do, does, either, else, ever, every,*
> *for, from, get, got, had, has, have, he, her, hers, him,*
> *his, how, however, i, if, in, into, is, it, its, just,*
> *least, let, like, likely, may, me, might, most, must, my,*
> *neither, no, nor, not, of, off, often, on, only, or, other,*
> *our, own, rather, said, say, says, she, should, since, so,*
> *some, than, that, the, their, them, then, there, these, they,*
> *this, tis, to, too, twas, us, wants, was, we, were, what,*
> *when, where, which, while, who, whom, why, will, with, would,*
> *yet, you, your, 's, n't, 'd, 'm, 'll, 've*

A future version may provide for user-specified stop words.

CoVec also ignores words that are not included in the vector set, and it displays a warning message every time it does so. That is, it treats the text as if the missing words and stop words were not in it; they are removed before processing.

Output file

The output, shown in bright white on the screen, is easily redirected to a file that can be opened with Excel or R. Just add **`> filename.txt`** (with any filename you want) to the command arguments.

Word sequence analysis

Analysis option **`-wordseq`** takes one or more files containing series of words, such as the results from a semantic fluency task, and calculates the mean similarity between consecutive words in each file. For example, the list of animals

Test1.txt: *cats dogs bears foxes giraffe rhino lion tiger armadillo pangolin*

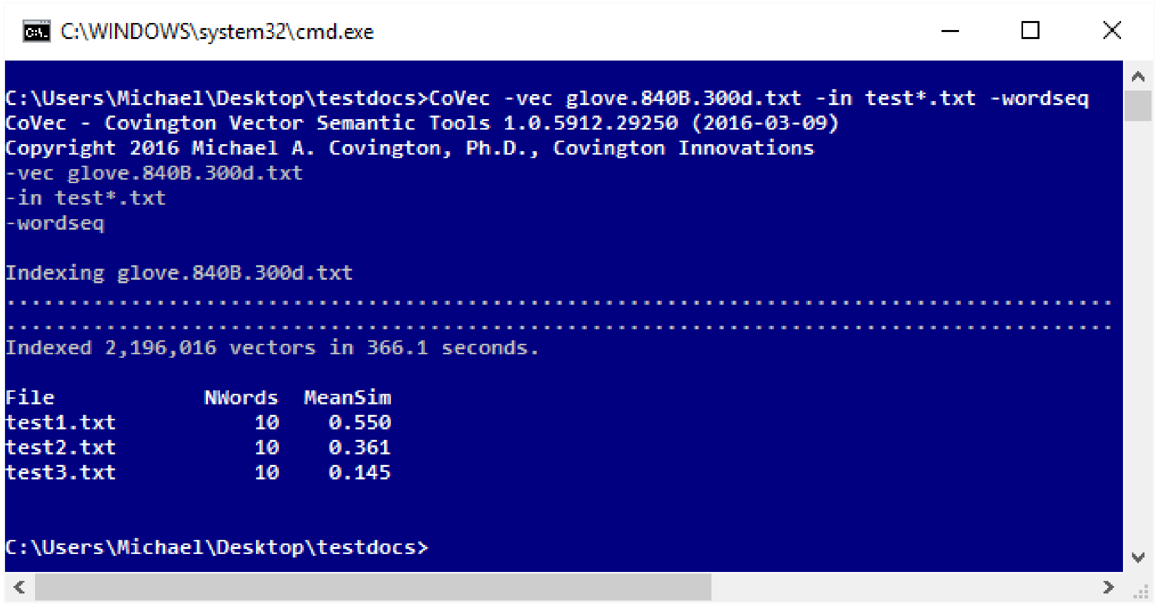
(from a real experiment) mostly has simliar or associated animals together, while

Test2.txt: *cats pangolin lion dogs armadillo bears tiger foxes giraffe rhino*

names the same animals in random order, and

Test3.txt: *cats theories geometry president watercolor flower election nebula telescope dubious*

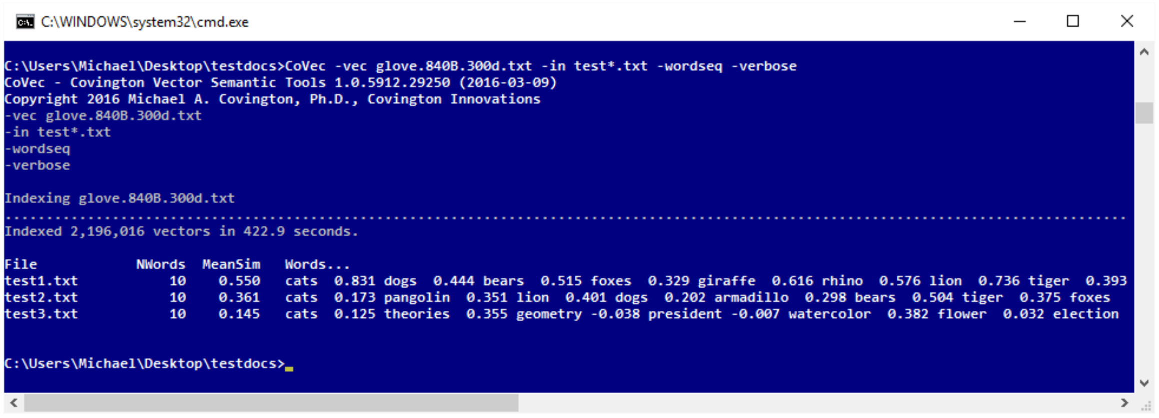
is a list of things that mostly have no similarity at all. Here's what it looks like to run CoVec on them (assuming all the files are in the same folder):



```
C:\Users\Michael\Desktop\testdocs>CoVec -vec glove.840B.300d.txt -in test*.txt -wordseq
CoVec - Covington Vector Semantic Tools 1.0.5912.29250 (2016-03-09)
Copyright 2016 Michael A. Covington, Ph.D., Covington Innovations
-vec glove.840B.300d.txt
-in test*.txt
-wordseq

Indexing glove.840B.300d.txt
........................................................................................
........................................................................................
Indexed 2,196,016 vectors in 366.1 seconds.

File            NWords  MeanSim
test1.txt           10    0.550
test2.txt           10    0.361
test3.txt           10    0.145


C:\Users\Michael\Desktop\testdocs>
```

Note that indexing the vector file takes several minutes; dots are displayed periodically while this is being done.

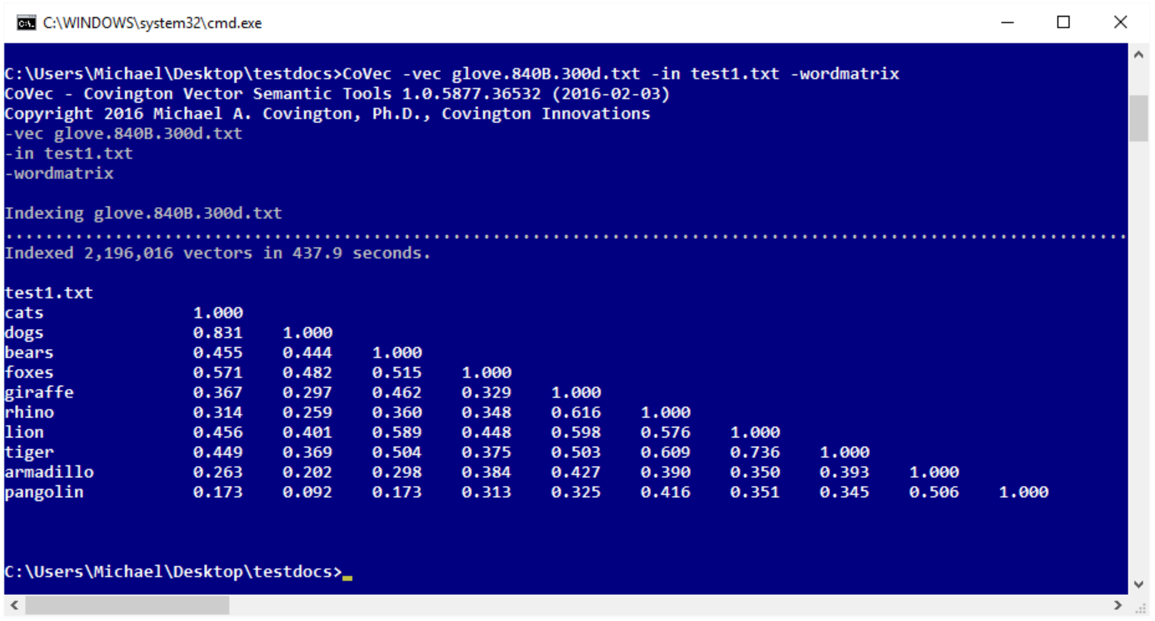
As expected, text 1 has a high score (0.550), text 2 has an intermediate score (0.361), and text 3 has a low score (0.145). With the **`-verbose`** option turned on, we also get the individual words and, between each pair of words, the word-to-word similarities:



```
C:\Users\Michael\Desktop\testdocs>CoVec -vec glove.840B.300d.txt -in test*.txt -wordseq -verbose
CoVec - Covington Vector Semantic Tools 1.0.5912.29250 (2016-03-09)
Copyright 2016 Michael A. Covington, Ph.D., Covington Innovations
-vec glove.840B.300d.txt
-in test*.txt
-wordseq
-verbose

Indexing glove.840B.300d.txt
.............................................................................................................................
Indexed 2,196,016 vectors in 422.9 seconds.

File            NWords  MeanSim   Words...
test1.txt           10    0.550   cats  0.831 dogs  0.444 bears  0.515 foxes  0.329 giraffe  0.616 rhino  0.576 lion  0.736 tiger  0.393
test2.txt           10    0.361   cats  0.173 pangolin  0.351 lion  0.401 dogs  0.202 armadillo  0.298 bears  0.504 tiger  0.375 foxes
test3.txt           10    0.145   cats  0.125 theories  0.355 geometry -0.038 president -0.007 watercolor  0.382 flower  0.032 election


C:\Users\Michael\Desktop\testdocs>_
```

Word matrix analysis

Analysis option **`-wordmatrix`** takes one or more files containing series of words and calculates the similarity between each word and each of the other words, displayed as a half-matrix. Unlike word sequence analysis, word matrix analysis does not care about the order in which the words are given, except that that is the order in which they are displayed. Here is an example, analyzing just one file:



```
C:\Users\Michael\Desktop\testdocs>CoVec -vec glove.840B.300d.txt -in test1.txt -wordmatrix
CoVec - Covington Vector Semantic Tools 1.0.5877.36532 (2016-02-03)
Copyright 2016 Michael A. Covington, Ph.D., Covington Innovations
-vec glove.840B.300d.txt
-in test1.txt
-wordmatrix

Indexing glove.840B.300d.txt
.........................................................................................
Indexed 2,196,016 vectors in 437.9 seconds.

test1.txt
cats                 1.000
dogs                 0.831    1.000
bears                0.455    0.444    1.000
foxes                0.571    0.482    0.515    1.000
giraffe              0.367    0.297    0.462    0.329    1.000
rhino                0.314    0.259    0.360    0.348    0.616    1.000
lion                 0.456    0.401    0.589    0.448    0.598    0.576    1.000
tiger                0.449    0.369    0.504    0.375    0.503    0.609    0.736    1.000
armadillo            0.263    0.202    0.298    0.384    0.427    0.390    0.350    0.393    1.000
pangolin             0.173    0.092    0.173    0.313    0.325    0.416    0.351    0.345    0.506    1.000


C:\Users\Michael\Desktop\testdocs>_
```

The columns are in the same order as the rows but are not labeled because the labels might be long. The similarity of *armadillo* to *pangolin* is 0.506. The similarity of each word to itself is 1.0.

Moving-window coherence analysis

Analysis option **`-coherence n`** takes one or more files containing series of words and calculates the coherence of each. Here *n* is a number.

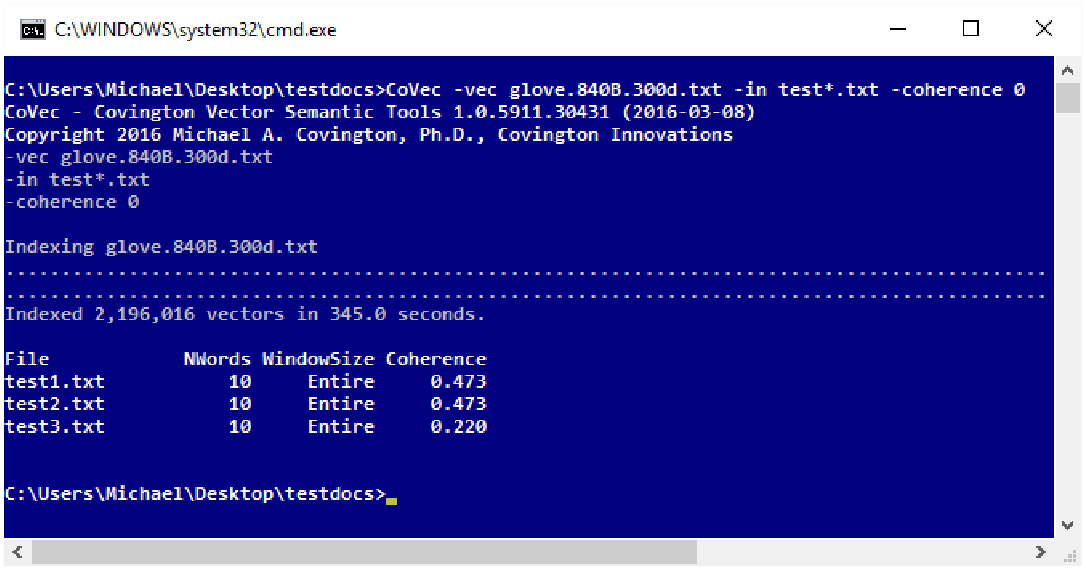
The coherence of a text is the average similarity of each of the words in each *n*-word segment of it to all the other words in the segment, regardless of where they occur. This is a measure of whether segments of that length commonly contain changes of topic.

Crucially, the text is not divided up into consecutive *n*-word segments. Instead, all positions of an *n*-word moving window are used (analogous to MATTR).[1] The results from all positions of the moving window are averaged.

If the specified window size is shorter than the text, the coherence is reported as NaN (not a number).

If the specified window size is 0, the entire text is used as one window. *Caution:* In this case, measured coherence reflects text length, lower for longer texts, in general, because of the greater likelihood of encountering a large shift somewhere in a longer text. The reason to use a fixed-length moving window is to make the measurement independent of text length.

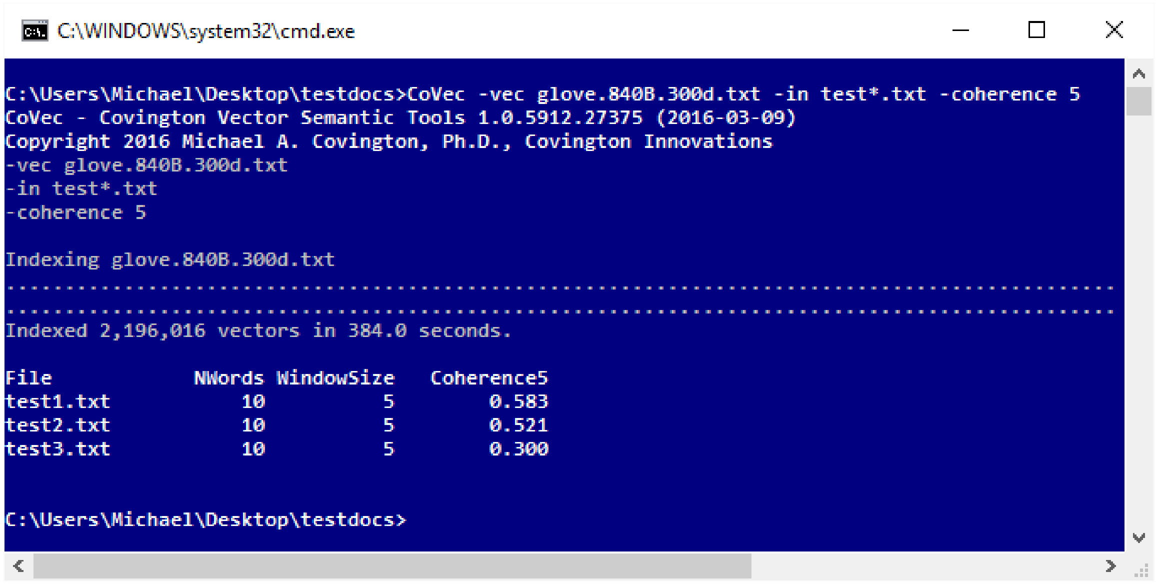
Looking back at our examples, the two lists of animals (texts 1 and 2) have equally high coherence, and the list of unrelated items has less coherence. Sure enough, that is what CoVec says, taking the lists whole (**`-coherence 0`**):


C:\WINDOWS\system32\cmd.exe

```
C:\Users\Michael\Desktop\testdocs>CoVec -vec glove.840B.300d.txt -in test*.txt -coherence 0
CoVec - Covington Vector Semantic Tools 1.0.5911.30431 (2016-03-08)
Copyright 2016 Michael A. Covington, Ph.D., Covington Innovations
-vec glove.840B.300d.txt
-in test*.txt
-coherence 0

Indexing glove.840B.300d.txt
................................................................................
................................................................................
Indexed 2,196,016 vectors in 345.0 seconds.

File              NWords WindowSize Coherence
test1.txt            10     Entire     0.473
test2.txt            10     Entire     0.473
test3.txt            10     Entire     0.220


C:\Users\Michael\Desktop\testdocs>_
```

[1] Covington, Michael A. and McFall, Joe D. (2010) Cutting the Gordian knot: the moving-average type–token ratio (MATTR). *Journal of Quantitative Linguistics* 17:94-100. Software available from www.ai.uga.edu/caspr.

The same is true, but a bit less dramatically, when a 5-word moving window is used, and in this case a small difference between Test 1 and Test 2 is picked up, because Test 2 jumps around more. Note that in this case the column is headed **`Coherence5`** so that, when imported into statistical, coherence measurements with different window sizes will have different names and can be compared.

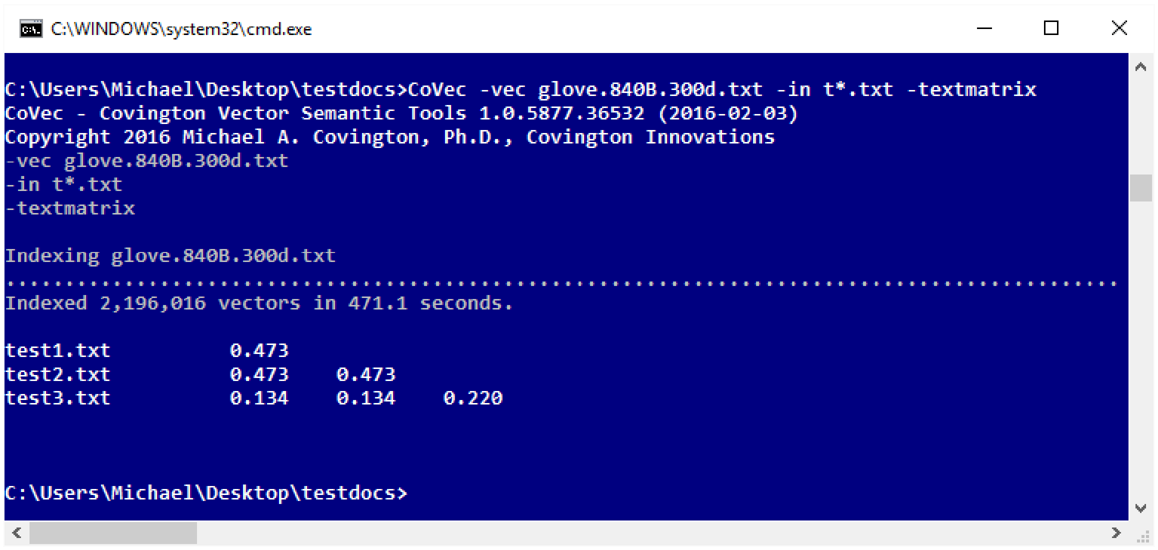


C:\WINDOWS\system32\cmd.exe

```
C:\Users\Michael\Desktop\testdocs>CoVec -vec glove.840B.300d.txt -in test*.txt -coherence 5
CoVec - Covington Vector Semantic Tools 1.0.5912.27375 (2016-03-09)
Copyright 2016 Michael A. Covington, Ph.D., Covington Innovations
-vec glove.840B.300d.txt
-in test*.txt
-coherence 5

Indexing glove.840B.300d.txt
............................................................................................
............................................................................................
Indexed 2,196,016 vectors in 384.0 seconds.

File            NWords WindowSize   Coherence5
test1.txt           10          5        0.583
test2.txt           10          5        0.521
test3.txt           10          5        0.300


C:\Users\Michael\Desktop\testdocs>
```

Text matrix analysis

Analysis option **`-textmatrix`** takes files containing series of words, such as text documents, and calculates the similarity of each document to each of the others, calculated as average similarity of each word in each document to each word in the other document. Note that the similarity of a document to itself is not 1.0; it is that document's coherence.



C:\WINDOWS\system32\cmd.exe

```
C:\Users\Michael\Desktop\testdocs>CoVec -vec glove.840B.300d.txt -in t*.txt -textmatrix
CoVec - Covington Vector Semantic Tools 1.0.5877.36532 (2016-02-03)
Copyright 2016 Michael A. Covington, Ph.D., Covington Innovations
-vec glove.840B.300d.txt
-in t*.txt
-textmatrix

Indexing glove.840B.300d.txt
..............................................................................................
Indexed 2,196,016 vectors in 471.1 seconds.

test1.txt          0.473
test2.txt          0.473    0.473
test3.txt          0.134    0.134    0.220



C:\Users\Michael\Desktop\testdocs>
```

Here documents 1 and 2 are essentially the same because they have exactly the same vocabulary, and document 3 is the odd one out. Naturally, the test would make more sense if one of the documents weren't just a scrambled version of another.

Clustering of the output of this analysis could be used for text classification.