

TermCluster Datapath and Tutorial

JJ O'Brien

770-235-6302

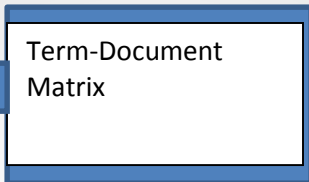
May 1, 2014



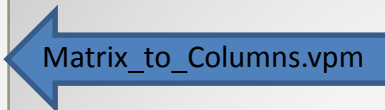
Raw VantagePoint
Phrase List



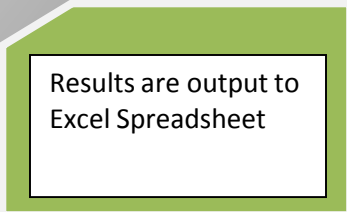
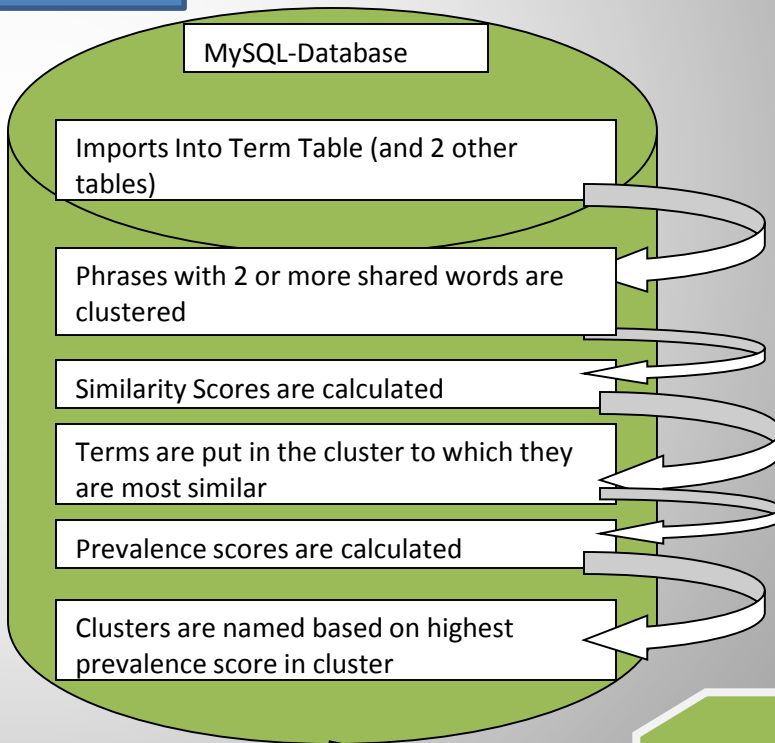
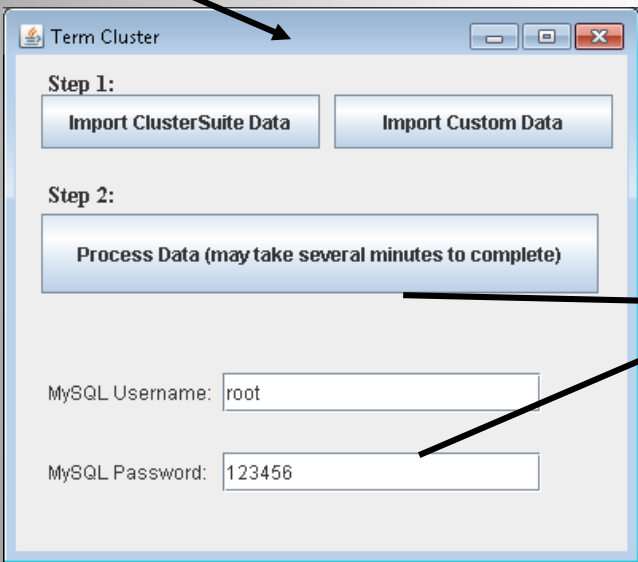
Cleaned VantagePoint
Phrase List



Term-Document
Matrix



Excel Spreadsheet
with keywords, ISI,
frequency



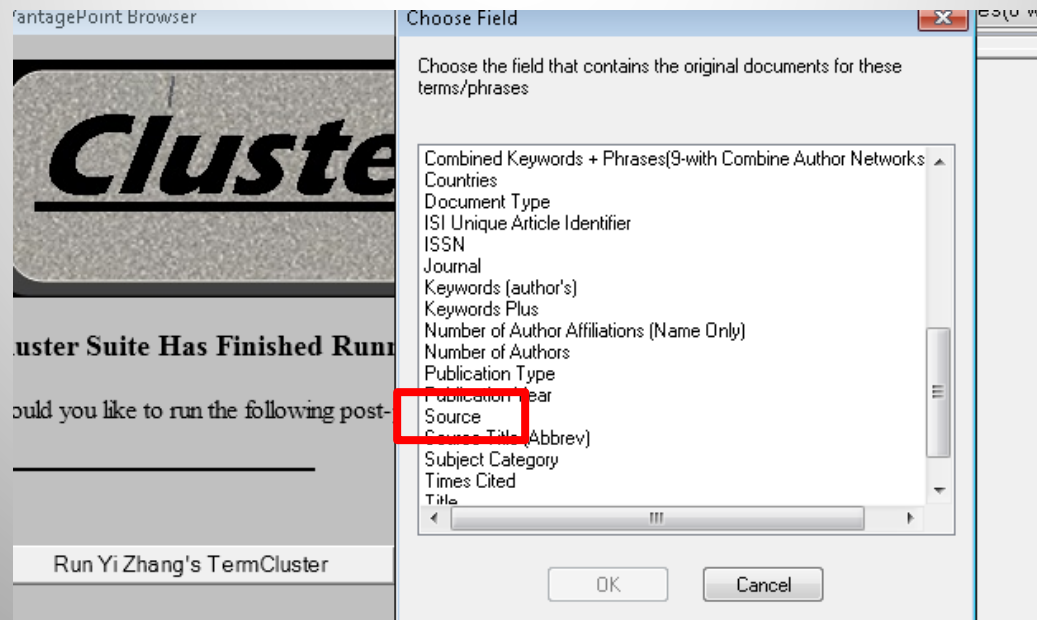
Results are output to
Excel Spreadsheet

MySQL

- TermCluster Requires MySQL to run. If you have not yet setup MySQL, please refer to the accompanying PowerPoint.

Create Input

- After ClusterSuite finishes running, you will be prompted to “Run Yi Zhang’s TermCluster”. If you select this option, you will immediately be prompted to select the field containing the documents which contained your keywords.
- In my example, this is the “Source” field.
- ClusterSuite then creates a Term-Document Matrix with the last field output from ClusterSuite and your document field.



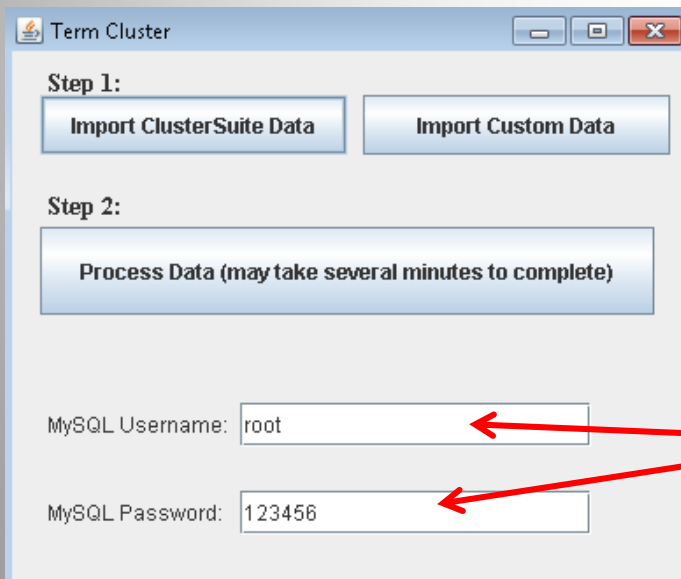
Matrix-To-Columns

- Next, ClusterSuite launches the Matrix_to_Columns.vpm and Matrix_to_Columns.xlsm combo to convert the Term-Document Matrix to a three-columned Excel file.
- Col A is the phrase from your ClusterSuite output. Col B is the document ISI or name. Col C is the number of times that phrase appears in that document.
- When it finishes, ClusterSuite will display a pop-up indicating the location of this file if you wish to view it (although this isn't necessary)

A	B	C	D
higher tumor uptake	3.21094E+11	1	
HUMAN INSULIN-RECEPTOR	3.26983E+11	1	
HUMAN INSULIN-RECEPTOR	3.28807E+11	1	
HUMAN TUMOR XENOGRAFT	2.80454E+11	1	
HUMAN TUMOR XENOGRAFT	3.17639E+11	1	
human umbilical vein endothelial cells HUVECs	2.36091E+11	1	
human umbilical vein endothelial cells HUVECs	2.70354E+11	1	
HYALURONIC-ACID	2.6709E+11	1	
HYALURONIC-ACID	2.7213E+11	1	
hydration	2.49138E+11	1	
hydration	2.57794E+11	1	
Hydrodynamic chromatography	2.72086E+11	1	
Hydrodynamic chromatography	2.86599E+11	1	
hydrolysis	2.47773E+11	1	
hydrolysis	2.72959E+11	1	
hydrophobic drug	2.58294E+11	1	
hydrophobic drug	2.62544E+11	1	
hydrophobic materials	2.32257E+11	1	
hydrophobic materials	2.36797E+11	1	

TermCluster Launch

- ClusterSuite now launches a runTermCluster.bat and runTermCluster.xlsm combo to launch termCluster.jar. This runs in the background, so don't worry about this step unless you are trying to edit the ClusterSuite code
- The termCluster GUI (Graphic User Interface) looks as follows.
- If termCluster is minimized for some reason when it first launches, look for the Java coffee cup icon on your task bar



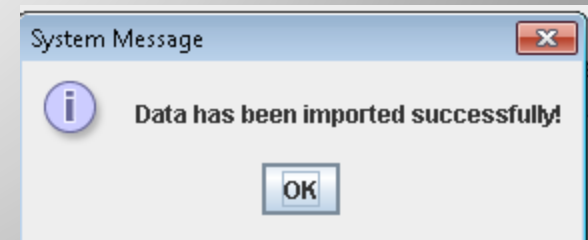
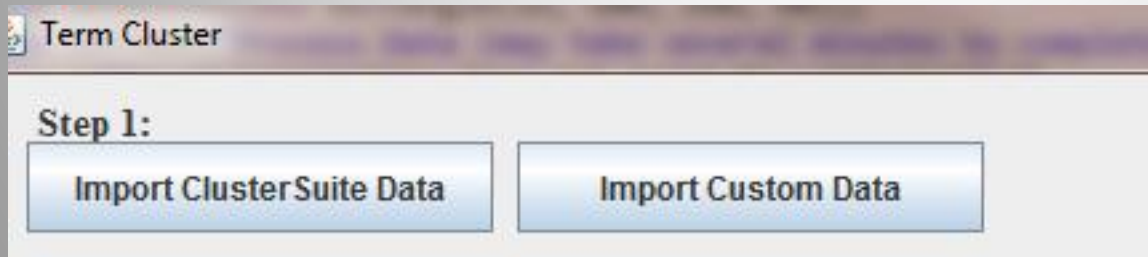
If you didn't use my recommended username and password when setting up your MySQL server, go ahead and type in your credentials before proceedings

TermCluster Objectives

- 1) Import Data into MySQL
- 2) Cluster phrases sharing two or more words together
- 3) Calculate average cosine Similarity scores for each phrase within each cluster to remove phrases from clusters to which they don't belong
- 4) For each phrase, pick one cluster to which they belong. Choose the cluster in which they have the highest similarity score.
 - For example if Phrase X is found in Cluster A and Cluster B, assume that Phrase X has an average similarity score of 0.7 across Cluster B's terms and 0.8 across Cluster A's terms. Phrase X will be removed from Cluster B.
- 5) Calculate Prevalence scores to determine what name to use for the Cluster. Imagine that Cluster A now consists of Phrase X (Prevalence 0.9) and Phrase Y (Prevalence 0.5). The new Cluster name for Cluster A will be Phrase X
- 6) Export all the clusters with their correct cluster names

Step 1: Import Data

- You have two options to import your data: (1) Either you can import the data that ClusterSuite output as an Excel file in Slide 5, or you can use your own data that uses the same Three-Column format as ClusterSuite's file (Phrase, ISI, frequency)
- Depending on your file size, this step may take several minutes to complete. You will see a message box indicating when the import is complete.



For Developers: Behind the Scenes of the Import (Ignore if you are not writing SQL or Java code for TermCluster)

(1) Insert phrases into term table. Insert number of words in each phrase as NumWord and assign a unique ID number

A	B	C	D
higher tumor uptake	3.21094E+11	1	
HUMAN INSULIN-RECEPTOR	3.26983E+11	1	
HUMAN INSULIN-RECEPTOR	3.28807E+11	1	
HUMAN TUMOR XENOGRAFT	2.80454E+11	1	
HUMAN TUMOR XENOGRAFT	3.17639E+11	1	
human umbilical vein endothelial cells HUVECs	2.36091E+11	1	
human umbilical vein endothelial cells HUVECs	2.70354E+11	1	
HYALURONIC-ACID	2.6709E+11	1	
HYALURONIC-ACID	2.7213E+11	1	
hydration	2.49138E+11	1	
hydration	2.57794E+11	1	
Hydrodynamic chromatography	2.72086E+11	1	
Hydrodynamic chromatography	2.86599E+11	1	
hydrolysis	2.47773E+11	1	
hydrolysis	2.72959E+11	1	
hydrophobic drug	2.58294E+11	1	
hydrophobic drug	2.62544E+11	1	
hydrophobic materials	2.32257E+11	1	
hydrophobic materials	2.36797E+11	1	

id	Term	NumWord
4447	ALBUMIN NANOPARTICLES	2
3919	BIODEGRADABLE NANOPARTICLES	2
4139	BIODEGRADABLE POLYMERIC NANOPARTICLES	3
4269	CHITOSAN NANOPARTICLES	2
4271	COBALT FERRITE NANOPARTICLES	3
4150	COMPOSITE NANOPARTICLES	2
4559	CONJUGATED NANOPARTICLES	2
4152	CORE-SHELL NANOPARTICLES	2
4602	DNA-capped nanoparticles	2
4676	FE3O4 NANOPARTICLES	2
4304	FUNCTIONALIZED GOLD NANOPARTICLES	3
4699	GLYCOLY-MODIFIED GELATIN NANOPARTICLES	3
3895	GOLD NANOPARTICLES	2
4766	IR-loaded Lf-CD nanoparticles Lf-CD/IR	4
3909	IRON-OXIDE NANOPARTICLES	2
4785	Lf-CD nanoparticles	2

id	Termid	Word	Mark
1	3868	Combined	0
2	3868	Keywords	0
3	3868	+	0
4	3868	Phrases(8-with	0
5	3868	High	0
6	3868	Extremes	0
7	3868	Removed)(with	0
8	3868	AcronymID)	0
9	3869	NANOPARTICLES	0
10	3870	cyanide	0
11	3870	ion	0
12	3871	nanotechnology	0
13	3872	IN-VIVO	0
14	3873	Development	0
15	3874	DOXORUBICIN	0
16	3875	IN-VITRO	0

(3) Insert Frequency and ISI numbers as "Record"s in TermRecordMatrix table

id	Termid2	Record	Frequency
1	3868	ISI Unique Article Identifier	Recs
2	3869	188554100008	1
3	3869	207063500014	1
4	3869	207857807776	1
5	3869	208164701749	1
6	3869	208164706631	1
7	3869	223654800021	1
8	3869	226206000003	1
9	3869	227546800003	1
10	3869	228815200014	1
11	3869	228815200015	1
12	3869	229038700024	1
13	3869	232257100003	1
14	3869	233878600001	1
15	3869	234019000005	1
16	3869	235440400020	1

(2) Separate each phrase into its individual words in WordsInTerm table. TermId in WordsInTerm is the same as id in term table.

Step 2: Process Data

- Once your data has been imported, it is ready to be clustered
- Press the Process Data button.
- This may take several minutes to complete, especially with larger record sets. If a recordset seems like it is taking a particularly long time to process, you may want to leave it alone for a while. You may continue to use other programs on your computer while it processes.

Step 2:

Process Data (may take several minutes to complete)

Clustering Algorithm

- For more information on the Clustering Algorithm used by TermCluster, refer to pages 4-5 in Yi Zhang's publication:

Zhang, Y., et al. "How to Combine Term Clumping and Technology Roadmapping for Newly Emerging Science & Technology Competitive Intelligence: The Semantic TRIZ Tool and Case Study." 14th International Society of Scientometrics and Informetrics (ISSI) Conference Proceedings. 2013.

For Developers: Behind the Scenes of the Process Data (Ignore if you are not writing SQL or Java code for TermCluster)

All Steps are called from
getProcessData() in the
MainFrame class

#	Step	Java Class	Method	MySQL Stored Procedure
0	Since termCluster only processes phrases containing between 2 and 4 words, move all other phrases to the exclude table	API Action	Exclude()	
1	Match phrases with at least two shared words to form clusters	API Action	callPopulateCluster(3)	populateCluster3
2	Calculate average similarity scores across every phrase in a cluster for each phrase in the Cluster	API Action	Similarity()	
3	Make sure each phrase belongs to only one Cluster. Compare each average similarity score that a phrase got in each of its clusters. Remove that phrase from all clusters except the one in which it has the highest similarity score	API Action	Labeling()	
4	Determine the Prevalence score for each cluster	API Action	Prevalence()	
5	Export only the name of the phrase with the highest prevalence score for each cluster, as this becomes the Cluster name	ReadMySQL	writeClusters()	
6	Delete everything from the termcluster database	API Action	cleanUp()	cleanup

For Developers: Notes I took on Similarity Calculation (Ignore if you are not writing SQL or Java code for TermCluster). See Yi Zhang's paper (Slide 11) for a much more detailed explanation

I started out with the following terms:

ID	Term	Number of Words	Publication A	Publication B
8517	Nano solar cell	3	3	1
8518	solar cell	2	1	1
8519	Nano cell	2	2	0

Step1) I'm only going to calculate the "nano solar cell" Group Name. Find all terms that have at least two words in common with "nano solar cell":

Group (Cluster) Name	Term ID
Nano solar cell	8517 (nano solar cell)
nano solar cell	8518 (solar cell)
Nano solar cell	8519 (Nano cell)

Step2) For each TermID in the Group Name, calculate the average similarity score across the other terms in the GroupName. For example, assume the following cosine similarity scores. I calculated these by hand and checked them with <http://www.appliedsoftwaredesign.com/archives/cosine-similarity-calculator>:

nano solar cell & nano solar cell = 1
nano solar cell & solar cell = .894
nano solar cell & nano cell = 0.949

Average Cosine Similarity: $(1 + 0.894 + 0.949)/3 = 0.948$

Next slide...

For Developers: Notes I took on Similarity Calculation Continued... (Ignore if you are not writing SQL or Java code for TermCluster). See Yi Zhang's paper (Slide 11) for a much more detailed explanation

Thus, we can update the Group Name Table as such...

Group (Cluster) Name	Term ID	Average Cosine Similarity
Nano Solar Cell	8517 (nano solar cell)	0.948
Nano Solar Cell	8518 (solar cell)	?
Nano Solar Cell	8519 (nano cell)	?

Step 3) Repeat, Step 2 for each remaining TermID:

solar cell & nano solar cell = 0.894

solar cell & solar cell = 1

solar cell & nano cell = 0.707

$(1 + 0.894 + 0.707)/3 = 0.867$

nano cell & nano solar cell = 0.949

nano cell & solar cell = 0.707

nano cell & nano cell = 1

$(0.949 + 0.707 + 1)/3 = 0.885$

Thus, our completed table is:

Now, if a term (we'll use solar cell as an example) is in more than one cluster, we have to find its average cosine similarity in each cluster. "Solar Cell" then remains with the cluster in which it has the highest similarity score

Group (Cluster) Name	Term ID	Average Cosine Similarity
Nano Solar Cell	8517 (nano solar cell)	0.948
Nano Solar Cell	8518 (solar cell)	0.867
Nano Solar Cell	8519 (nano cell)	0.885

For Developers: Data flow through MySQL components used by ClusterSuite (Ignore if you are not writing SQL or Java code for TermCluster)

Import Data

Component	Type	Columns	Action Performed	Description
Term	Table	Id, Term, NumWord	Populated	Starting point for all phrases
termrecordmatrix	Table	Id, Termid2, Record, Frequency	Populated	Stores publication information for each phrase
WordsInTerm	Table	Id, termid, Word, Mark	Populated, words in phrases counted	Each word of each phrase is listed individually here
Excluded	Table	Id, Term, NumWord	Populated	Stores phrases with fewer than 2 words or more than 4 words

Populate Cluster 3

Component	Type	Columns	Action Performed	Description
populateCluster3	Stored Procedure	--	Executes; Also uses the countWordsinCommon function	this procedure populates cluster3 with all terms from term table that-- have at least 2 words in common
Cluster3	Table	id, GroupName, Termid4, NumSharedWords, Prevalence	Populated	Holds the cluster names in the Group Name column and the terms in the termid4 column
Term	Table	Id, Term, NumWord	Used to populate Cluster3	Starting point for all phrases
WordsInTerm	Table	Id, termid, Word, Mark	Used to determine which prases have words in common	Each word of each phrase is listed individually here

Next Slide...

For Developers: Data flow through MySQL components used by ClusterSuite Continued... (Ignore if you are not writing SQL or Java code for TermCluster)

Similarity

Component	Type	Columns	Action Performed	Description
Cluster3	Table	id, GroupName, Termid4, NumSharedWords, Prevalence	Used in calculation	Holds the cluster names in the Group Name column and the terms in the termid4 column
Termrecordmatrix	Table	Id, Termid2, Record, Frequency	Used to deal with the pieces of the calculation that use document data	Stores publication information for each phrase
Similarity	Table	Id, GroupName, Termid1, Similarity	Populated	Contains Similarity scores

Labeling

Component	Type	Columns	Action Performed	Description
Cluster3	Table	id, GroupName, Termid4, NumSharedWords, Prevalence	Phrases are only allowed in one cluster	Holds the cluster names in the Group Name column and the terms in the termid4 column
Similarity	Table	Id, GroupName, Termid1, Similarity	Used to determine which phrases to remove from Cluster3	Contains Similarity scores

Next Slide...

For Developers: Data flow through MySQL components used by ClusterSuite Continued... (Ignore if you are not writing SQL or Java code for TermCluster)

Prevalence

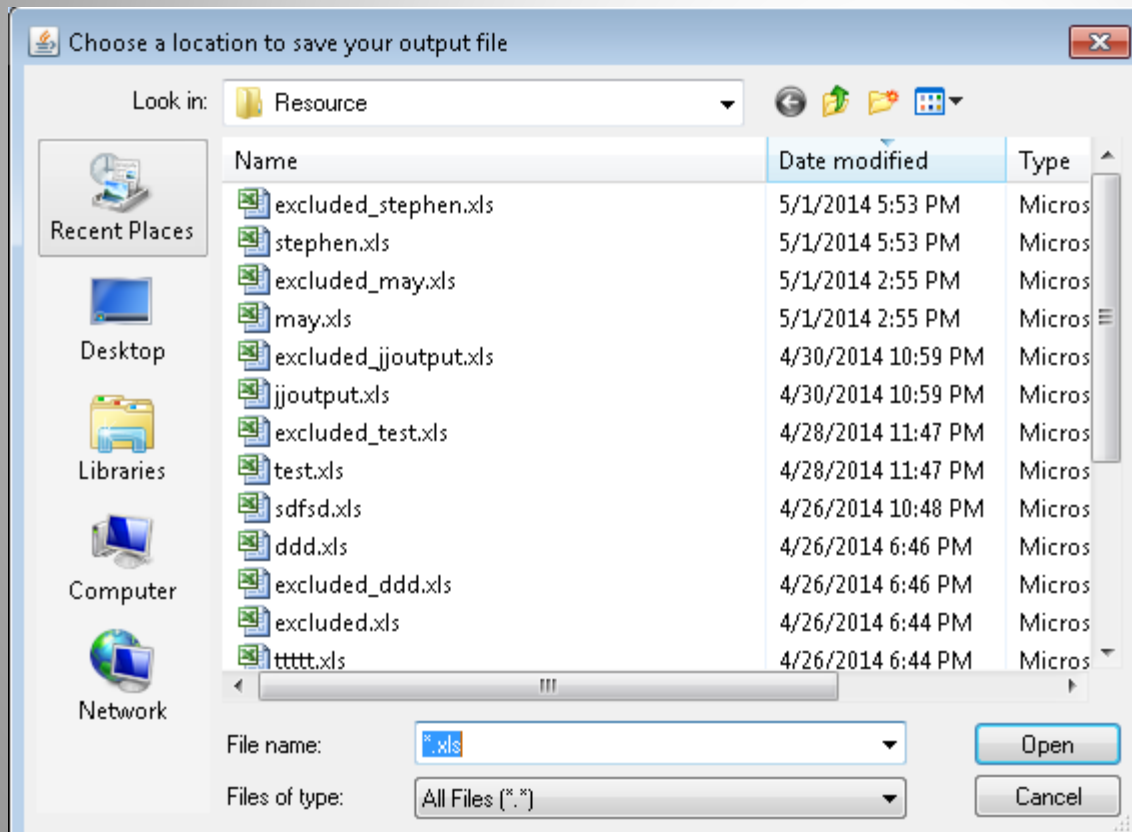
Component	Type	Columns	Action Performed	Description
Cluster3	Table	id, GroupName, Termid4, NumSharedWords, Prevalence	Prevalence Score is Calculated	Holds the cluster names in the Group Name column and the terms in the termid4 column
Termrecordmatrix	Table	Id, Termid2, Record, Frequency	Used in calculating prevalence score	Stores publication information for each phrase

WriteClusters

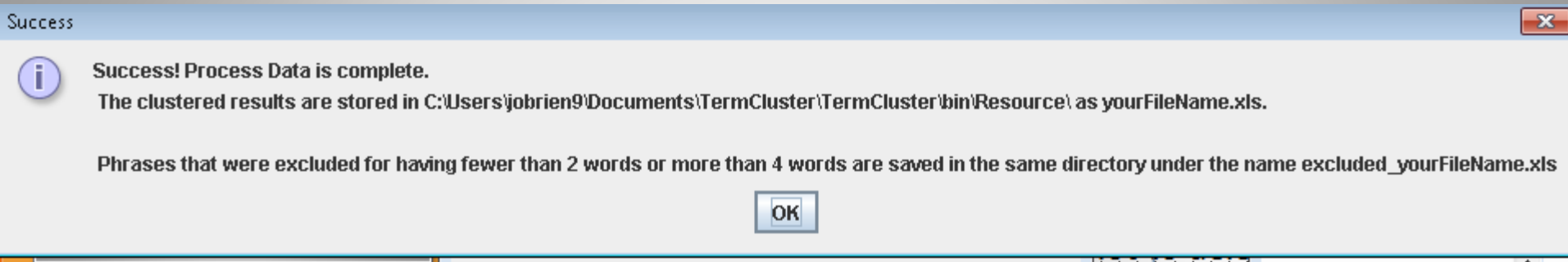
Component	Type	Columns	Action Performed	Description
Cluster3	Table	id, GroupName, Termid4, NumSharedWords, Prevalence	Clusters are pulled from here	Holds the cluster names in the Group Name column and the terms in the termid4 column
Term	Table	Id, Term, NumWord	Used to get name of phrases which are represented by a termid4 number in Cluster3	Starting point for all phrases
Excluded	Table	Id, Term, NumWord	Pulled to their own Excel spreadsheet	Stores phrases with less than 2 words or more than 4 words

Save Clusters

- When Step 2 finishes running, it will prompt you to save your output files. Name your output whatever you like and be sure not to delete the “.xls” at the end!



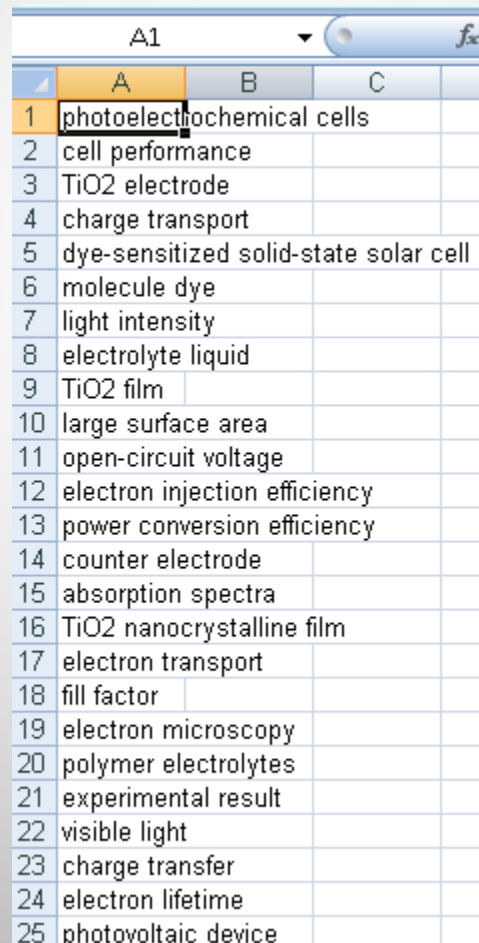
View Results



- Next, TermCluster will display a pop-up box that shows the location of the results file.
- For your convenience, the file location will also be opened in Windows Explorer.
- Last, note that the excluded_yourFileName.xls is also saved to the same location as the results files. This contains any phrases that contain fewer than 2 words or more than 4 words and were, therefore, unable to be processed by TermCluster.

TermCluster Complete

- Success! Your clusters should be visible in Column A



A screenshot of an Excel spreadsheet showing a list of terms in column A. The spreadsheet has columns labeled A, B, and C, and rows numbered 1 through 25. The terms listed in column A are:

	A	B	C
1	photoelectrochemical cells		
2	cell performance		
3	TiO2 electrode		
4	charge transport		
5	dye-sensitized solid-state solar cell		
6	molecule dye		
7	light intensity		
8	electrolyte liquid		
9	TiO2 film		
10	large surface area		
11	open-circuit voltage		
12	electron injection efficiency		
13	power conversion efficiency		
14	counter electrode		
15	absorption spectra		
16	TiO2 nanocrystalline film		
17	electron transport		
18	fill factor		
19	electron microscopy		
20	polymer electrolytes		
21	experimental result		
22	visible light		
23	charge transfer		
24	electron lifetime		
25	photovoltaic device		