Supplemental Materials

In this document we provide step by step instructions on how to perform the analyses discussed in the paper using TACIT. All the analyses were performed using TACIT version Beta 2.

1. Senate Crawler. The Senate crawler was used to compile a dataset of transcript of speeches given in the Senate floor between Sep 10, 1996 and Sep 10, 2001, and also between Sep 12, 2001 and Sep 12, 2006. We will use this dataset to demonstrate the utility of each plugin, and will provide step by step instructions on how to use them.

Given that we are interested in periods that extend beyond one Congress, we choose All Congresses (1989-2016) , but also specified a data range (e.g. 9/10/1996-9/10/2001). For our analyses, we decided to limit the records per congress member to five random transcripts (Figure 1a). We ran the crawl once for date range of 9/10/1996-9/10/2001, and once for 9/12/2001-9/12/2006.

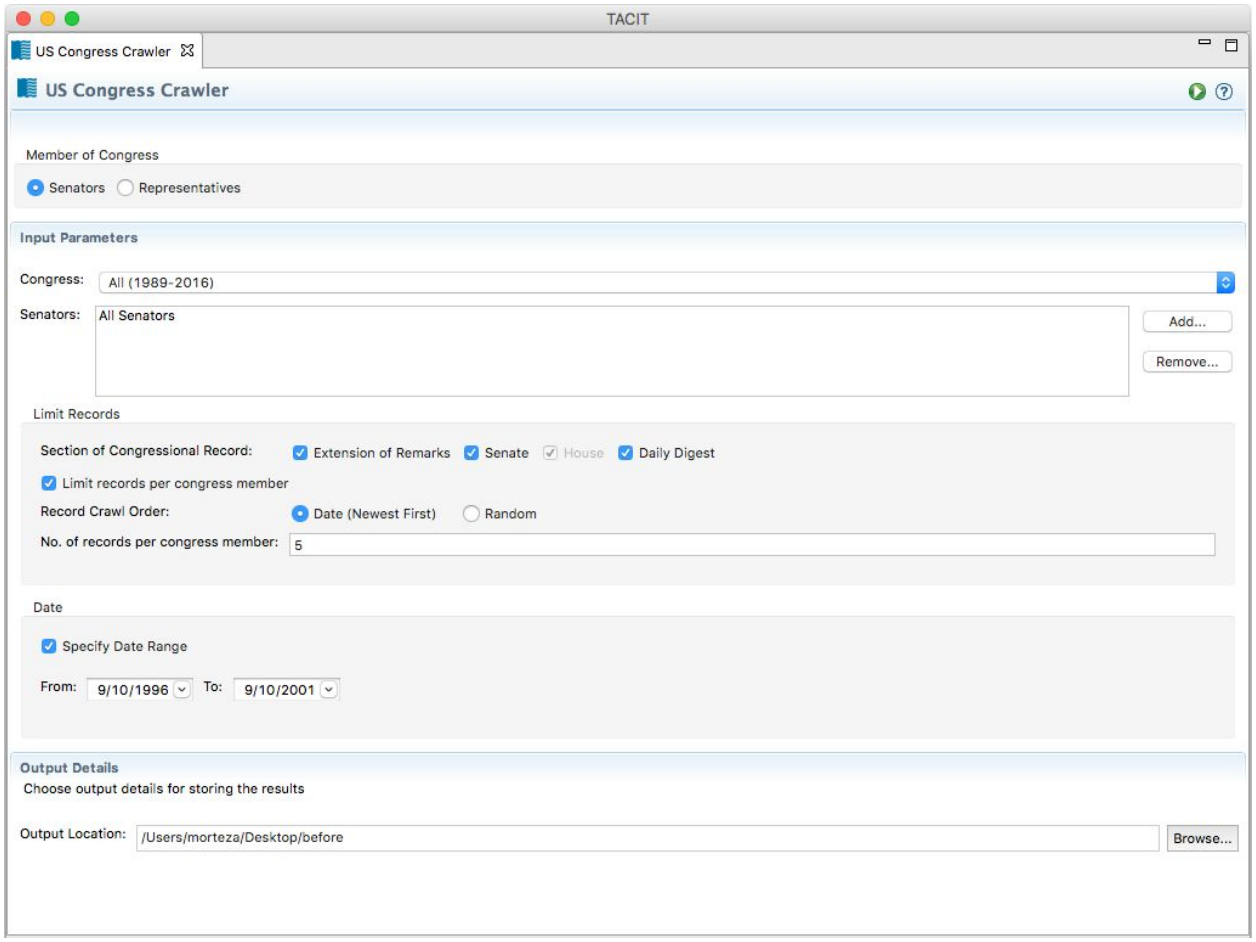

Figure 1a: Running the senate crawler to collect transcript of speeches given between 9/10/1996-9/10/2001. The above setting limits the number of transcripts per Congress member to five.

Please note that the crawler saves a CSV file named "congress-crawler-summary-DATE" which includes a summary of the data crawled. This summary includes information such as the Senators' name, political affiliation, State, Congress number, date of the speech given, title of the speech and name of the file that the speech is saved under. If the parent directory of the crawled

text is chosen as input to another plugin, this CSV file should be moved from the directory. Otherwise, the CSV file will get included as part of the analysis.

2. LIWC Word Count. We used TACIT's LIWC word count plugin to investigate the frequency of words related to social processes (e.g., talk, share, friends) and moral rhetoric of loyalty to ingroup members. The first analysis was done using the LIWC 2007 dictionary (Pennebaker et al., 2007), and the second using the Moral Foundations Dictionary (Graham, Haidt, & Nosek, 2009). We ran the analysis once for the speeches given prior to Sep 11 attacks, and once for the ones given after the attacks. After the word count terminates, the results can be imported into a statistical analysis software for further processing. The hypothesis and the results of our analysis are discussed in the main body of the paper.

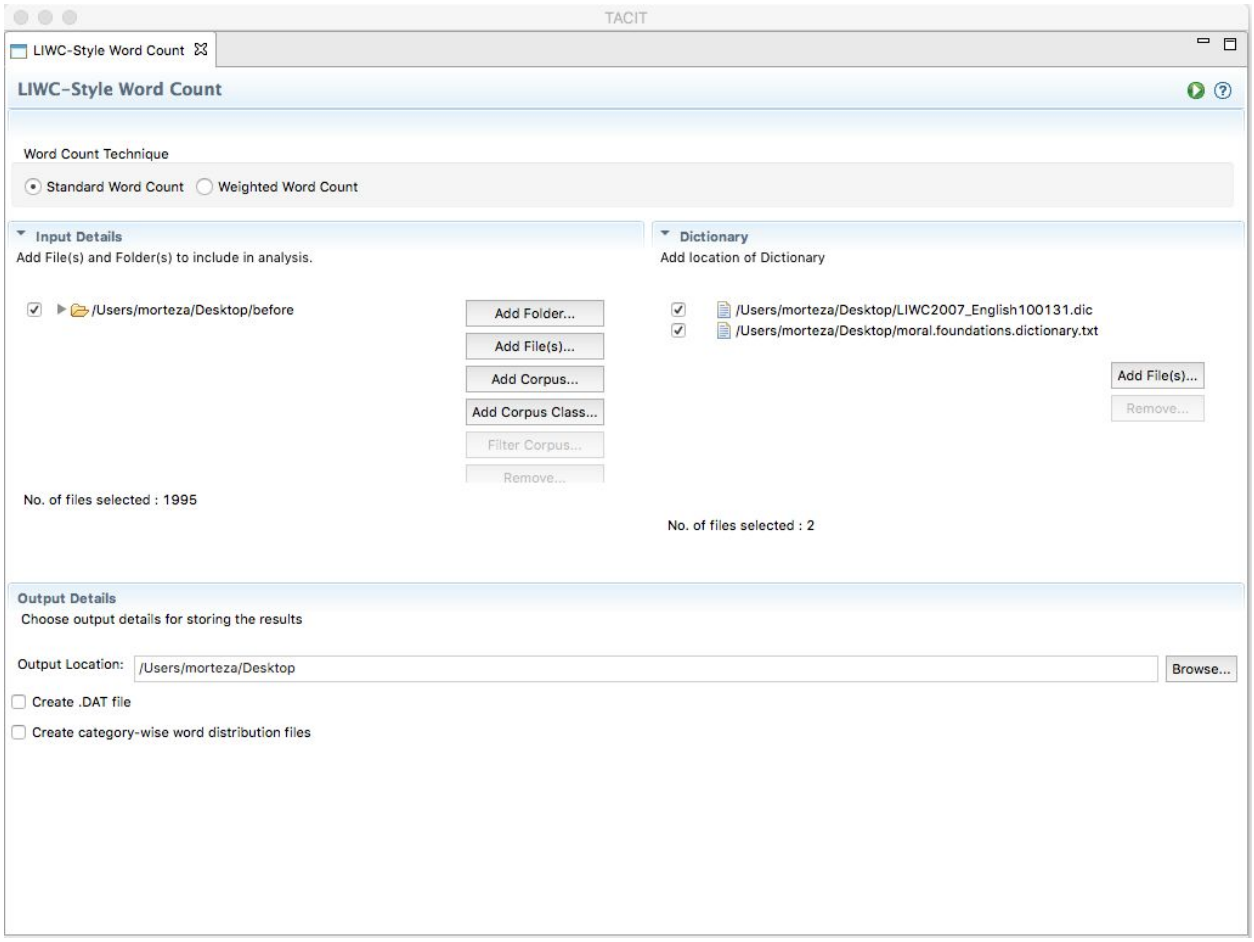

Figure 2a: Running the LIWC-Style Word Count plugin on the transcript of speeches given prior to Sep 11 attacks. Here we are using both the LIWC 2007 dictionary and the Moral Foundations Dictionary to analyze the documents.

3. Co-Occurrence Analysis: We used the co-occurrence analysis plugin to investigate how often two sets of words co-occur with one another in the pre and post datasets. These words are saved in a text file. The seed file needs to be a plain text file. The first seed file includes the following words:

Clinton Lewinsky affair

And the second file includes the following:

Iraq war wmd

Please note that with pre-processing set to on, the case of the words do not affect the analysis, as all words are converted to lowercase (by default). For this analysis, we set the size of the moving window (neighborhood size) to 10 words, and the threshold to 2. The threshold sets the minimum number of seed words that need to co-occur in order for that instance to be considered as a co-occurrence hit. This plugin returns two files by default: 1. co-occur_phrases_DATE.csv: lists the seed combinations found, the file names the seed combinations occurred in, the line number and the specific phrase that the seeds co-occurred in 2. co-occur_seedfrequencies_DATE.csv: lists every seed combination, with minimum words in each combination set to the threshold, and the co-occurrence count of each of the combinations. Additionally, if the "Build co-occurrence matrices" is selected in the interface, then the complete matrix of co-occurrence of all the words in the input is returned. As discussed in the paper, we ran this analysis once for the pre-9/11 dataset and once for the post-9/11 transcripts. We found that the first set of words co-occur 25 times in the pre-9/11 dataset, and do not co-occur in the post. However, the second set co-occur 41 times in the post dataset, and three times in the pre.

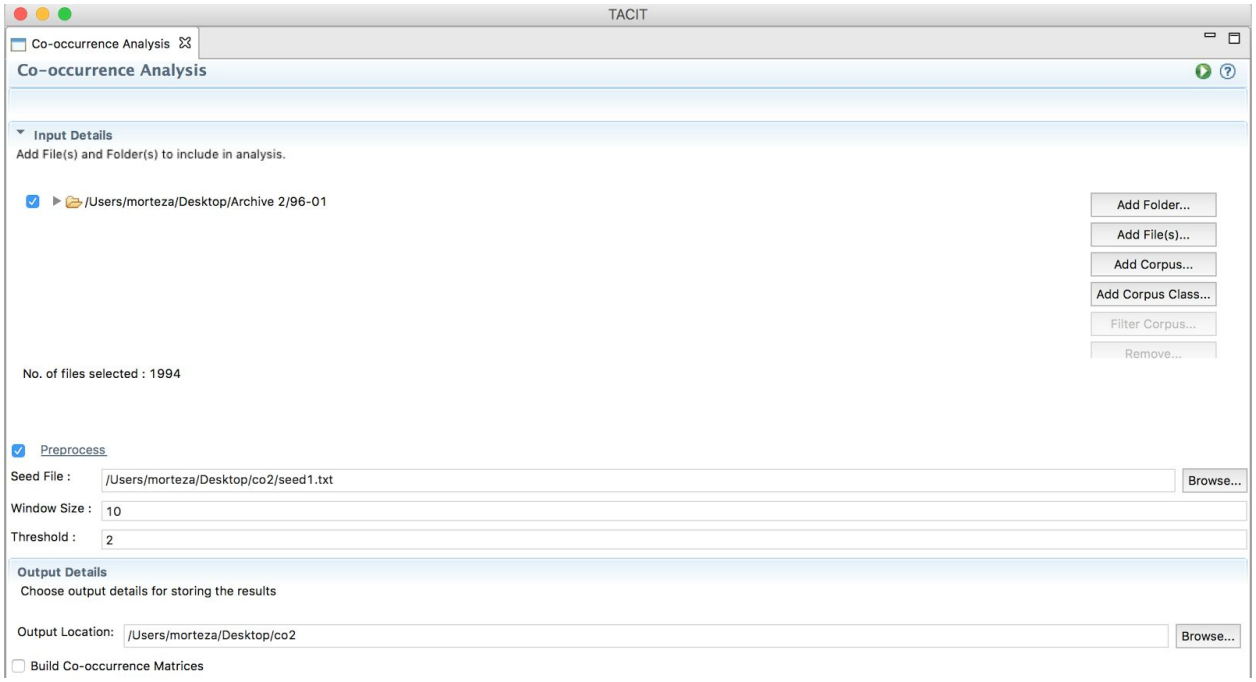

Figure 3a: Running the Co-Occurrence plugin on the transcript of speeches. Here we are setting the size of the moving window (neighborhood size) to 10 words, and the threshold to 2 (at least two of the seed words need to co-occur in order for the instance to be considered a hit).

4. Support Vector Machine Classification. We ran SVM classification on the transcripts to see if we can reliably distinguish between speeches given prior to 9/11 to those given post 9/11. Setting *k* to 10 results in the plugin training a classifier on 90% of the data and testing on the other 10%, and repeating this process 10 times. This plugin returns a file named SVM-Classification-DATE.csv, which includes the classification accuracy of each fold, and *k* files named SVM-Classification-weights-kn-DATE (with 1 <= n =< k). These files include the weight of each of the words (features) in the files. The higher the absolute value of the weight of a word, the more significant of a role that word had in that classification.

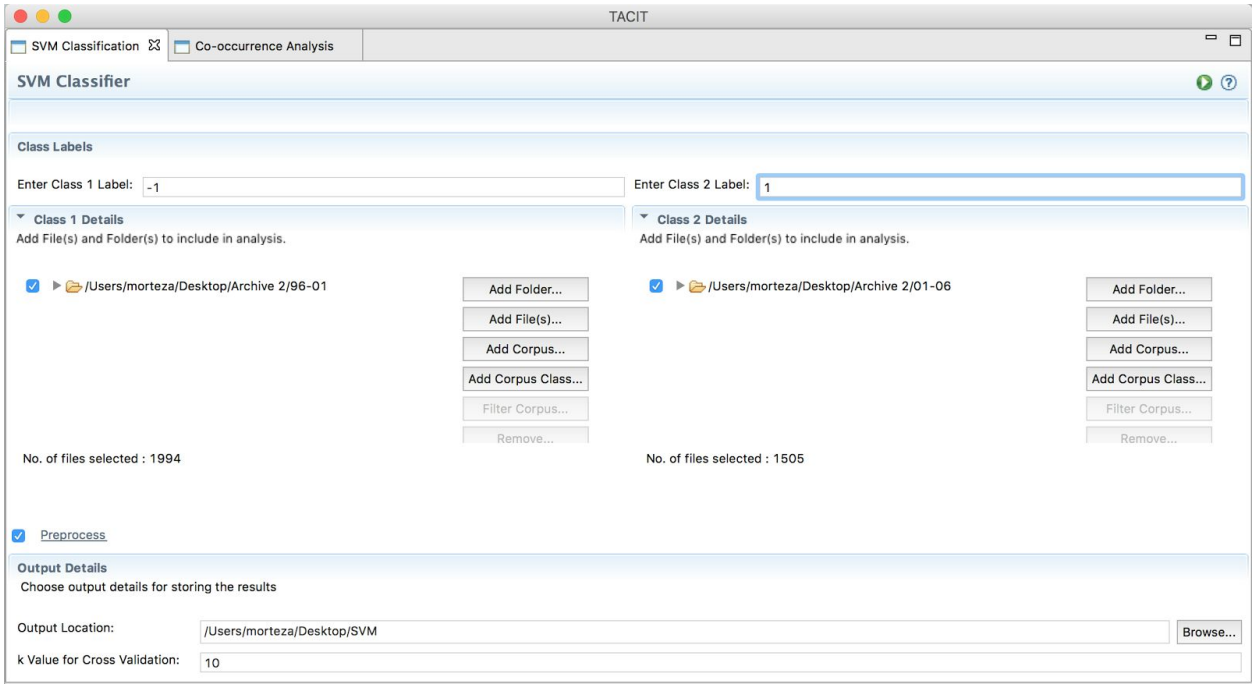

Figure 4a: Running the SVM plugin to classify the pre and post speeches. The labels can be any number, and *k* can be any value larger than 0.

5. Naive Bayes Classification. In order to demonstrate the benefits of the Naive Bayes plugin, we downloaded another set of transcripts of Senate speeches given between 10/13/06 to 10/14/11. We then used this plugin to perform a three-way classification of the transcripts. Similar to the SVM plugin, number of folds for cross-validation needs to be set to a number greater than 0. Unlike the SVM plugin, Naive Bayes does not return any files. However, several measures of accuracy of classification is returned by the plugin.

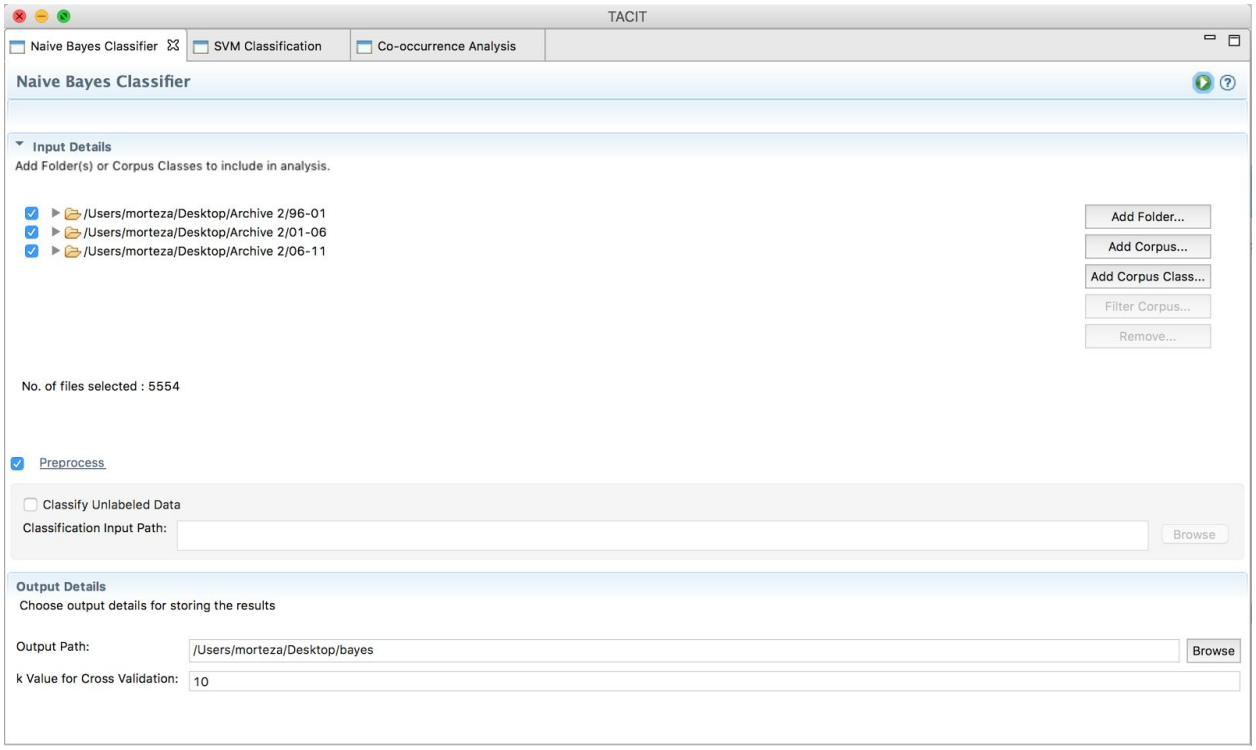

Figure 5a: Running a three-way classification using the Naive-Bayes plugin.

6. KMeans Clustering. As mentioned in the paper, k-means is an unsupervised algorithm for clustering data into groups. When k-means is used, usually ground truth is not known, and we would want to know the groups of texts that go together better than others. This plugin was used to perform unsupervised clustering on the pre and the post transcripts (combined together), to see if two coherent sets of clusters would emerge, representing the pre and the post groups. Therefore, we set k to 2, and ran the plugin. The KMeans plugin returns a text file which includes the group membership for each input file.

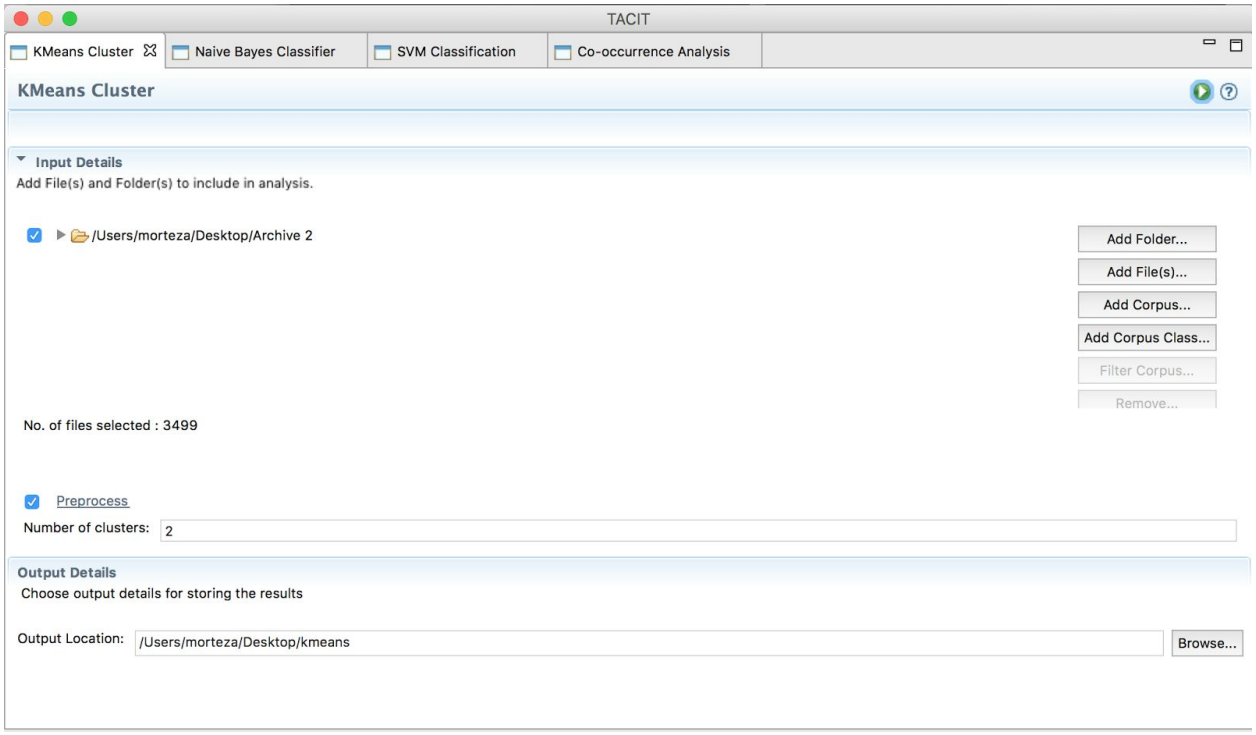

Figure 6a: Running unsupervised clustering using the KMeans plugin, with k set to 2.

7. Latent Dirichlet allocation: We did topic-modeling to investigate the differences in the topics that exist in the pre and the post dataset. This plugin returns several files: 1. lda.topic-composition-DATE: includes the probability of each topic per document. 2. lda.topic-composition: lists the topic composition of each document 3. lda.topic-keys-DATE: includes the composition of each topic 4. lda.word-weights-DATE: includes the weight of each word per topic. This analysis revealed that although there are several topics that are repeated in the pre and the post dataset, the post dataset includes topics about Afghanistan, Iraq and Katrina which are not discussed in the pre dataset.

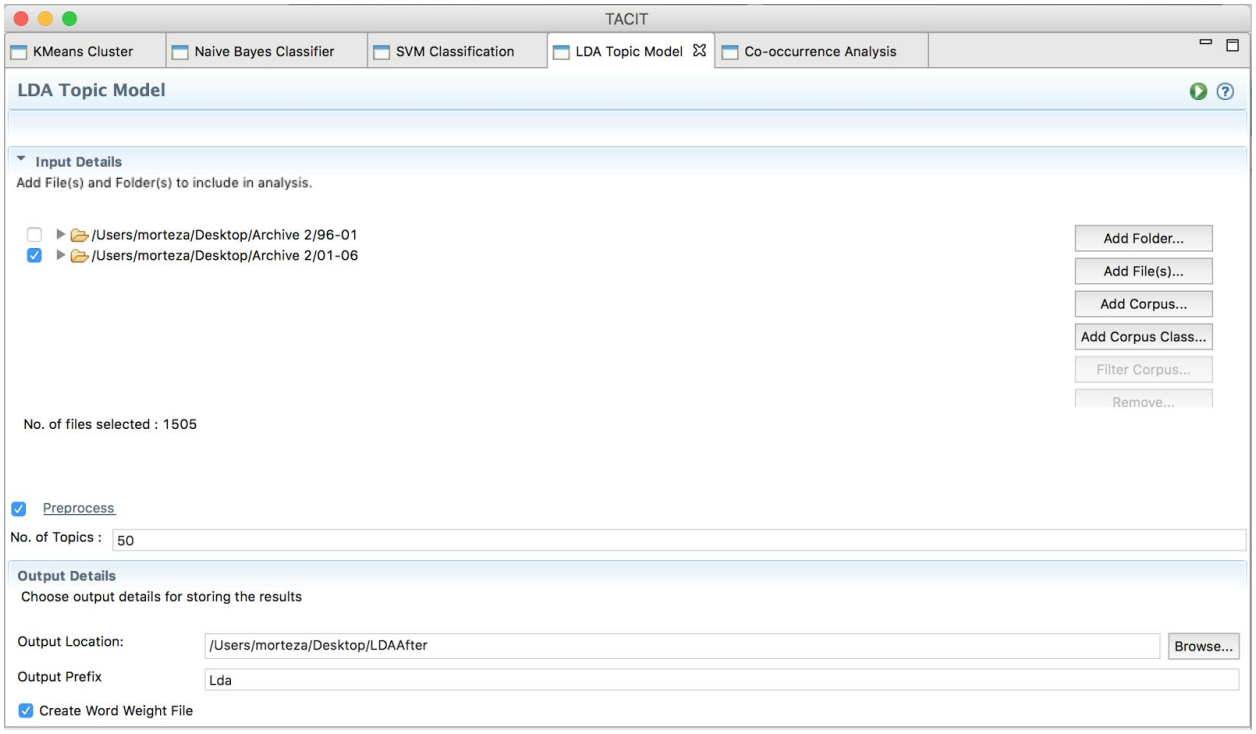

Figure 7a: Running LDA Topic Modeling with the number of topics set to 50.

8. Z-Label LDA: Running z-label LDA in TACIT is very similar to running LDA, with the exception that in z-label a seed file needs to be provided. The seed file should be in plain text, with words in line n being seed words for topic n. Similar to LDA, z-label returns several files containing information about the topic model: 1. wordsinTopics(Phi)-DATE.csv: lists the probability of each word for each topic 2. topicwords-DATE: has the composition of each topic, and the weight of each word under that topic 3. topicsPerDocument(theta)-DATE: includes the topic composition of each document.

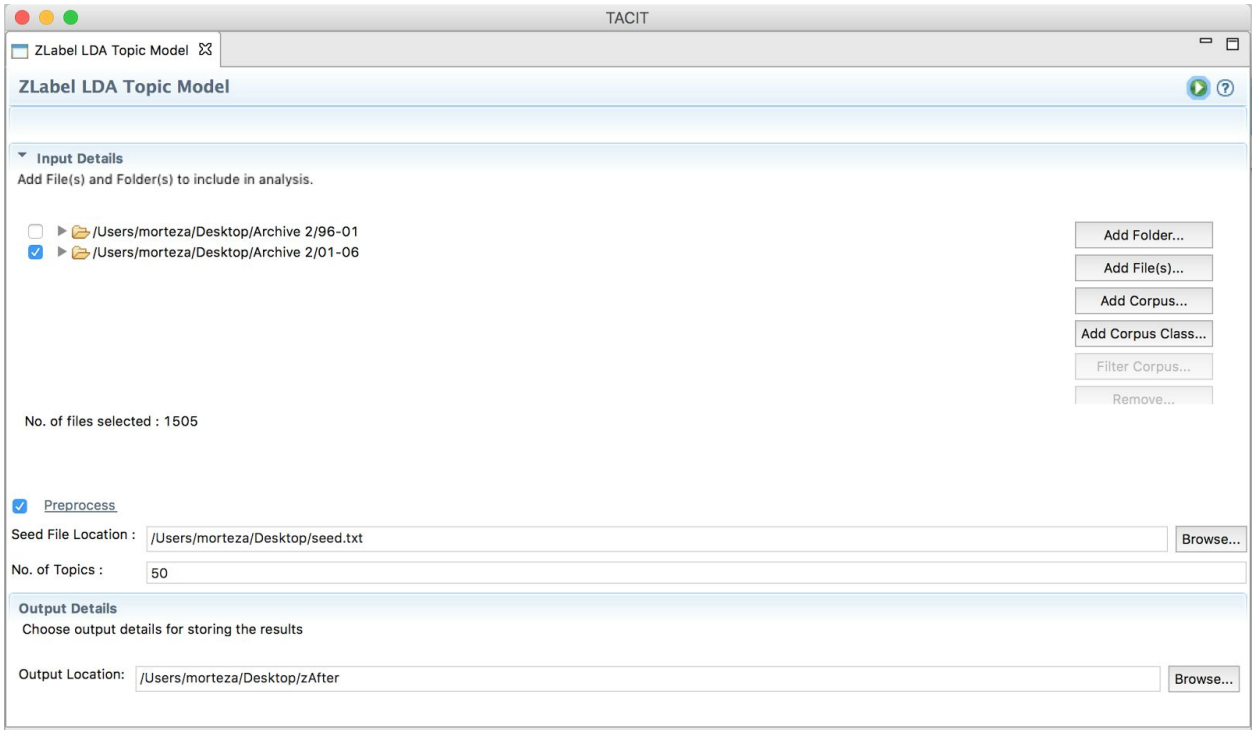

Figure 8a: Running z-label LDA Topic Modeling with the number of topics set to 50.