

CMSC 409: Artificial Intelligence
Fall 2019, Instructor: Dr. Milos Manic, <http://www.people.vcu.edu/~mmanic>
Project 4

CMSC 409: Artificial Intelligence

Project No. 4

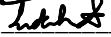
Due: Nov. 12, 2019, noon

Student certification:

Team member 1:

Print Name: Judah Sebastian *Date:* 11/12/2019

I have contributed by doing the following: I wrote the combine_stemmed_words function, and helped with project comprehension

Signed:  (you can sign/scan or use e-signature)

Team member 2:

Print Name: Gavin Alberghini *Date:* 11/12/2019

I have contributed by doing the following: Assisted with code, Answered questions, and Assisted with data output

Signed: Gavin Alberghini (you can sign/scan or use e-signature)

Team member 3:

Print Name: Michael Poblacion *Date:* 11/12/2019

I have contributed by doing the following: Implemented pre-processing and clustering algorithms

Signed: Michael Poblacion (you can sign/scan or use e-signature)

Pr.4.

1. Download and unzip “Project4_sentences.zip” and “Project4_code.zip” files.

A set of sentences is given in the file “sentences.txt”. Each sentence is a line in the file. Create the feature vector by writing a program that applies the following text mining techniques to this set of sentences.

- A. Tokenize sentences
- B. Remove punctuation and special characters
- C. Remove numbers
- D. Convert upper-case to lower-case
- E. Remove stop words. A set of stop words is provided in the file “stop_words.txt”
- F. Perform stemming. Use the Porter stemming code provided in the file “Porter_Stemmer_X.txt”
- G. Combine stemmed words.
- H. Extract most frequent words.

Provide the feature vector in your report.

Note:

The feature vector contains unique sets of words that appear in the set of sentences provided.

The file “Project4_code.zip” contains implementations of the Porter Stemmer in several languages. You can use any version of the code provided (provided versions of the code are Java, Matlab, Python, and C). Make sure you rename your file accordingly. More source code for the Porter Stemmer can be found here: <http://tartarus.org/martin/PorterStemmer/>

Assignment 4

CMSC 409

11/08/19

Gavin Alberghini, Michael Poblacion, Judah Sebastian

How many clusters/topics have you identified?

- We identified 10 clusters

A. Tokenize sentences

- *We tokenize sentences in order to discover our feature space, which consists of unique words.*
- *At this step in the process, we don't "lose" anything in terms of context*

B. Remove punctuation and special characters

- *We do this because punctuation and special characters are not valuable as cluster features for the goals of this assignment.*
- *We lose sentence context, tone and certain descriptors of words*

C. Remove numbers

- *We remove numbers because they are quantifiers they have no value in determining the topic of a cluster.*
- *We lose descriptors in terms of quantity, and lose more sentence context*

D. Convert upper-case to lower-case

- *This is purely for formatting within our program*
- *We lose nothing in terms of content, but we lose certain grammar contexts like the start of a sentence and proper nouns*

E. Remove stop words. A set of stop words is provided in the file "stop_words.txt"

- *These words are prepositions that are descriptors and articles, and have no value in determining the topic of a cluster.*
- *We lose descriptors and sentence context*

F. Perform stemming. Use the Porter stemming code provided in the file "Porter_Stemmer_X.txt"

- *We perform stemming in order to reduce similar words such as "running," "runner" and "runs" to the same word base, or stem. It consolidates our column space without loss of word meaning.*
- *We lose sentence context and pieces of words*

G. Combine stemmed words.

- *We combine stemmed words to remove duplicates from the feature vector, in order to consolidate our column space.*
- *We lose sentence context*

H. Extract most frequent words.

- *We extract most frequent words in order to identify the common topics that are likely to form clusters.*
- *We lose words from our feature vector*

What drives the dimensionality of the TDM?

- Rows : Number of sentences
- Columns : Number of unique words
- Overall dimensionality is then driven by the size of the document (number of sentences) and the diversity of the vocabulary (number of unique words).

What can you do to reduce that dimensionality?

- You can potentially reduce the column space by trimming the amount of "unique" words that are considered for each sentence. Alternatively you can decide to only consider sentences containing a certain amount of words.

Does the order of data being fed to algorithm matter?

- Yes, the order of feeding the data affects the growth/mutation of the clusters when using the FCAN algorithm.

['autonom', 'sedan', 'road', 'mile', 'per', 'hour', 'machin', 'learn', 'artifici', 'intellig', 'car', 'kilomet', 'home', 'bedroom', 'bath', 'live', 'room', 'larg', 'kitchen', 'size', 'around', 'interior', 'possibl', 'iot', 'devic', 'includ', 'light', 'system', 'hous', 'come', 'park', 'space', 'us', 'two', 'sens', 'on', 'design', 'over', 'updat', 'comput', 'understand', 'major', 'thing', 'vehicl', 'function', 'person']

Our program outputs our TDM and clusters in the console.

Results

28 Sentences in cluster 1

[0.17857142857142858, 0.14285714285714285, 0.17857142857142858, 0.35714285714285715, 0.25000000000000006, 0.21428571428571427, 0.03571428571428571, 0.0, 0.0, 0.0, 0.21428571428571427, 0.21428571428571427, 0.0, 0.07142857142857142, 0.0, 0.03571428571428571, 0.0, 0.0, 0.10714285714285714, 0.0, 0.10714285714285714, 0.14285714285714285, 0.07142857142857142, 0.07142857142857142, 0.07142857142857142, 0.03571428571428571, 0.03571428571428571, 0.03571428571428571, 0.07142857142857142, 0.0, 0.03571428571428571, 0.14285714285714285, 0.07142857142857142, 0.07142857142857142, 0.03571428571428571, 0.07142857142857142, 0.14285714285714285, 0.10714285714285714, 0.10714285714285714, 0.0, 0.03571428571428571, 0.07142857142857142, 0.10714285714285714, 0.03571428571428571, 0.10714285714285714, 0.14285714285714285]

[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.8333333333333334, 0.5, 0.6666666666666666, 1.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.16666666666666666, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.16666666666666666, 0.16666666666666666, 0.0, 0.0, 0.0, 0.0, 0.0]

[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.8, 1.4, 0.6, 0.4, 0.6, 0.6, 0.2, 0.2, 0.0, 0.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.2, 0.2, 0.0, 0.0, 0.2, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.2, 0.0, 0.0, 0.0, 0.0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[illegible]

[0, 0, 0, 0, 0, 0, 1, 2, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0]

[illegible][illegible]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 3, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]

Cluster 1 has 28 sentences, meaning there are 28 sentences that share a set of common topics.

Cluster 2 has 6 sentences, Cluster 3 has 5 sentences.

Since the rest of the clusters contain only one sentence, it means the remaining sentences do not share enough commonality in topic with the rest of the sentences.