Hons. Degree in Computing H4016 Text Mining & Information Retrieval

Unit 4 – Generating test and training datasets

Ref for this section: *Konchady, Text Mining Application Programming, Thomson 2006, chapter 4*

Context

To-date we have covered the theory relevant to steps 1, 2 and 3. This Unit looks at practical issues arising from the application of that theory.

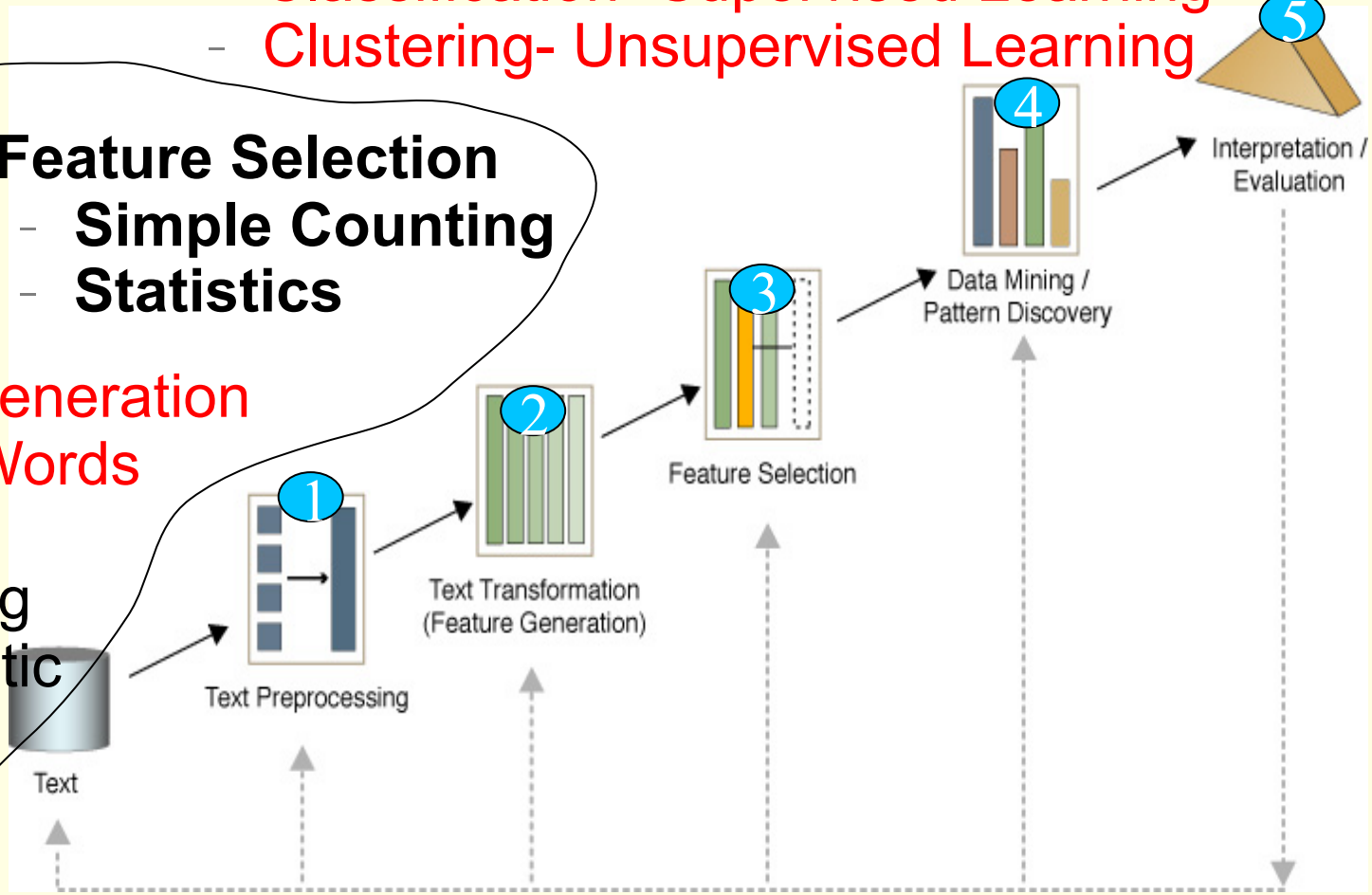
5. Analyzing Results

4. Text/Data Mining
   - Classification- Supervised Learning
   - Clustering- Unsupervised Learning

3. **Feature Selection**
   - **Simple Counting**
   - **Statistics**

2. Features Generation
   - Bag of Words

1. Text Preprocessing
   - Syntactic/Semantic Text Analysis



Objectives

- Understand the relationship between a training process and a test process

The following exercise appears in lab sheet #3:

Below are two processes, one to prepare data for training a classification model; the other is to prepare data to test that model.

- Where there are differences between the two processes, explain if that will effect the results of testing the model?
- If the Dictionarystemmer was used in the test process rather than the training process, would that be OK?

Process for training dataset

- Process document (texts=../training; purnebelow=3; vector_creation=TFIDF;)
  - StringTokeniser
  - EnglishStopWordFilter
  - DictionaryStemmer
  - PortersStemmer
- Output wordlist

. . .generate model

Process for test datasets

- Root
  - Process document – create wordlist from saved file
  - Process document (texts=../test; purnebelow=-1; vector_creation=occurences; word_list=lab3.txt)
    - StringTokeniser
    - StopWordFilterFile
    - LovinsStemmer

. . . test model

Processes explained

- The training process uses the 15 texts from lab#2 to generate a decision tree that can predict if a text is about healthcare, crime or kenya based on keywords present in the 15 texts.
- The test process takes the model (e.g. decision tree) generated in the training process, and applies it to the 4 'unseen' documents. The decision tree will classify each one as being about either healthcare, crime or kenya .
- We can verify its accuracy based on how it classify's the 4 documents.

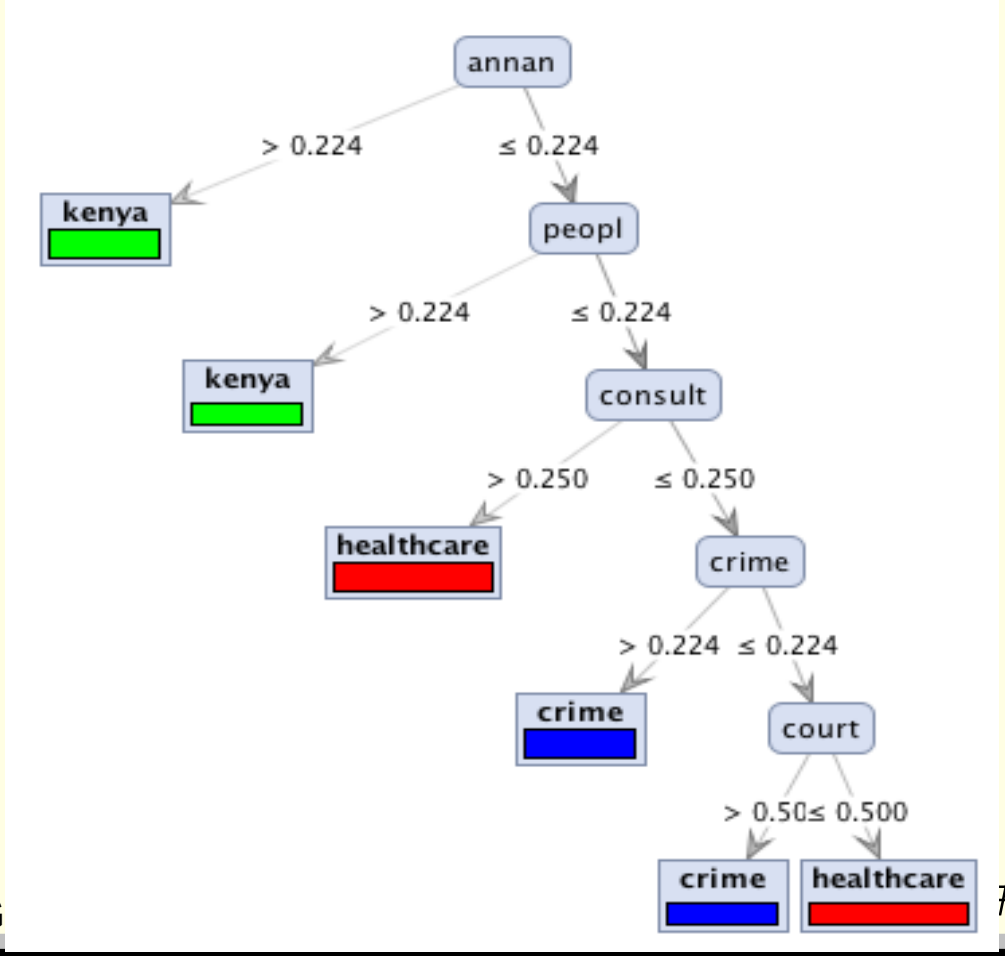
- Prunebelow=3 removes all term with less than three characters
- Prunebelow=-1 does not prune any terms

Output word list / input word list

- The final list of attributes resulting from the training process is as follows:

word
annan
clash
consult
court
crime
former
hospit
hse
kofi
peopl
rift
right
un
vallei
western

These are the attributes (column headings) that appear in the document vector.
Learners generated from this document vector will be based on some or all of these terms, e.g.

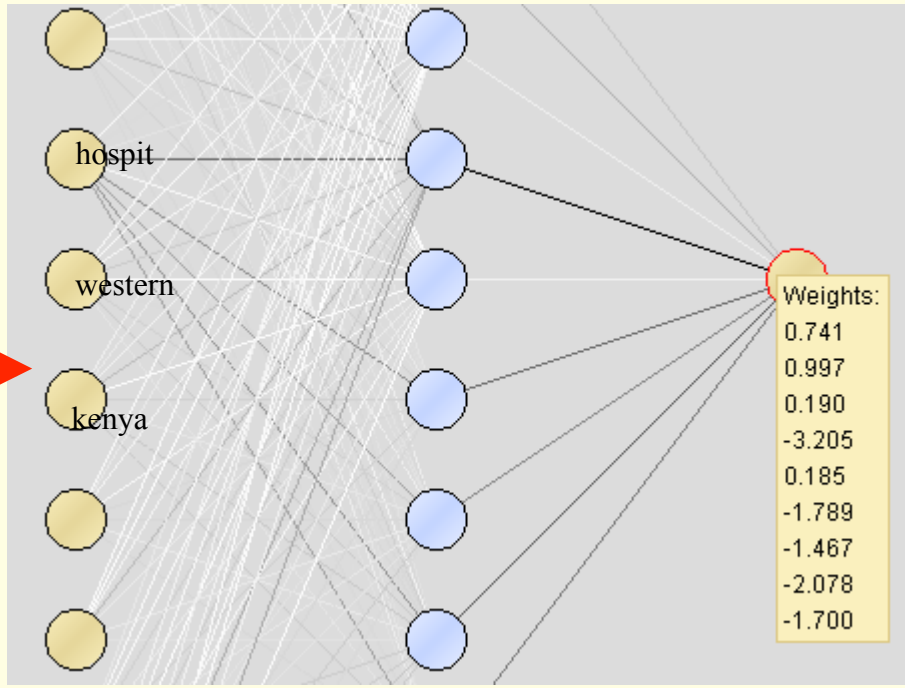


Output word list / input word list

- If the model (e.g. Decision tree on the last slide) is to be tested on documents not yet processed, the document vector generated from the test documents MUST have the same column headings as the document vector generated by the training data.
  - Why?

The decision tree on the last slide only used 5 of the 15 attributes. Other learners, such as k-NN or a neural network, would use all 15 attributes.

Neural network for the 15 documents. Darker links refer to more influential terms (high positive or high negative values)



Document vector of training dataset

annan	clash	consult	court	crime	former	hospit	hse	kofi	peopl	rift	right	un	vallei	western
0	0	0	0	0	0	0	0	0	0.500	0.500	0	0	0.500	0.500
0.459	0.229	0	0	0	0.229	0	0	0.229	0.229	0.459	0.229	0.229	0.459	0.229
0	0.447	0	0	0	0	0	0	0	0.447	0.447	0	0	0.447	0.447
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.447	0	0	0	0	0	0	0.894	0	0	0

Document vector of test dataset, without using an input wordlist

due	effort	execut	feud	former	fridai	galwai	gardai	gestur	hand	heroin	hse	huge	investig	issu
0	0	0.248	0	0	0	0	0	0	0	0	0.248	0.248	0	0.248
0	0.160	0	0.160	0.160	0	0	0	0.160	0.160	0	0	0	0	0
0	0	0	0	0	0	0.295	0	0	0	0	0	0	0.295	0
0.198	0	0	0	0	0.198	0	0.198	0	0	0.198	0	0	0	0

Document vector of test data set using the wordlist from the training process

annan	clash	consult	court	crime	former	hospit	hse	kofi	peopl	rift	right	un	vallei	western
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0.500	0	0	0	0	0.500	0	0	0.500	0	0	0	0.500	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	0

Vector Creation. . .

Diferrences in a Decision tree's rules for Vector creation=TFIDF versus Vector creation=occurrences

Occurrences

```
annan > 0.500: kenya {crime=0, kenya=3, healthcare=0}
annan ≤ 0.500
|   peopl > 0.500: kenya {crime=0, kenya=2, healthcare=0}
|   peopl ≤ 0.500
|   |   consult > 0.500: healthcare {crime=0, kenya=0, healthcare=3}
|   |   consult ≤ 0.500
|   |   |   crime > 0.500: crime {crime=3, kenya=0, healthcare=0}
|   |   |   crime ≤ 0.500
|   |   |   |   court > 0.500: crime {crime=2, kenya=0, healthcare=0}
|   |   |   |   court ≤ 0.500: healthcare {crime=0, kenya=0, healthcare=2}
```

TFIDF

```
annan > 0.224: kenya {crime=0, kenya=3, healthcare=0}
annan ≤ 0.224
|   peopl > 0.224: kenya {crime=0, kenya=2, healthcare=0}
|   peopl ≤ 0.224
|   |   consult > 0.250: healthcare {crime=0, kenya=0, healthcare=3}
|   |   consult ≤ 0.250
|   |   |   crime > 0.224: crime {crime=3, kenya=0, healthcare=0}
|   |   |   crime ≤ 0.224
|   |   |   |   court > 0.500: crime {crime=2, kenya=0, healthcare=0}
|   |   |   |   court ≤ 0.500: healthcare {crime=0, kenya=0, healthcare=2}
```

Remaining operators

- Tokenisers: What would be the impact of one process using an n-gram tokeniser?

- Filters: Does it matter that different filters were used?
  - Filter Stopword (English) ?
  - Filter Stopword (Dictionary) ?
  - Filter Tokens by Length?

- Stemmers: Does it matter that different stemmers were used?
  - DictionaryStemmer?
  - Lovins Stemmer? (keny, peopl, valle, hospit)
  - Porters Stemmer? (kenya, peopl, vallei, hospit)

Summary

- The document vector created during the training process MUST match the document vector created during the test process
  - Outputting the wordlist in the training process, and inputting the wordlist again in the test process will create identical columns headings
- Need to ensure tokens, and how they are counted also match in both preparation blocks.