Processing of large document collections

Part 1b (text representation, text categorization)

Helena Ahonen-Myka

Spring 2006

2. Text representation

- selection of terms
- vector model
- weighting (TF*IDF)

2

Text representation

- text cannot be directly interpreted by the many document processing applications
- we need a compact representation of the content
- which are the meaningful units of text?

3

Terms

- words
  - typical choice
  - set of words, bag of words
- phrases
  - syntactical phrases (e.g. noun phrases)
  - statistical phrases (e.g. frequent pairs of words)
  - usefulness not yet known?

4

Terms

- part of the text may not be considered as terms: these words can be removed
  - very common words (function words):
    - articles (a, the) , prepositions (of, in), conjunctions (and, or), adverbs (here, then)
  - numerals (30.9.2002, 2547)
- other preprocessing possible
  - stemming (recognization -> recogn), base words (skies -> sky)
- preprocessing depends on the application

5

Vector model

- a document is often represented as a vector
- the vector has as many dimensions as there are terms in the whole collection of documents

6

Vector model

- in our sample document collection, there are 118 words (terms)
- in alphabetical order, the list of terms starts with:
  - absorption
  - agriculture
  - anaemia
  - analyse
  - application
  - ...

7

Vector model

- each document can be represented by a vector of 118 dimensions
- we can think a document vector as an array of 118 elements, one for each term, indexed, e.g. 0-117

8

Vector model

- let d1 be the vector for document 1
- record only which terms occur in document:
  - d1[0] = 0 -- absorption doesn't occur
  - d1[1] = 0 -- agriculture -"-
  - d1[2] = 0 -- anaemia -"-
  - d1[3] = 0 -- analyse -"-
  - d1[4] = 1 -- application occurs
  - ...
  - d1[21] = 1 -- current occurs
  - ...

9

Weighting terms

- usually we want to say that some terms are more important (for some document) than the others -> weighting
- weights usually range between 0 and 1
  - 1 denotes presence, 0 absence of the term in the document

10

Weighting terms

- if a word occurs many times in a document, it may be more important
  - but what about very frequent words?
- often the TF*IDF function is used
  - higher weight, if the term occurs often in the document
  - lower weight, if the term occurs in many documents

11

Weighting terms: TF*IDF

- TF*IDF = term frequency * inversed document frequency
- weight of term t_k in document d_j:

$$tfidf(t_k, d_j) = \#(t_k, d_j) \cdot \log \frac{|Tr|}{|Tr(t_k)|}$$

- where
  - $\#(t_k, d_j)$: the number of times t_k occurs in d_j
  - Tr: the documents in the collection
  - $Tr(t_k)$: the documents in Tr in which t_k occurs

12

Weighting terms: TF*IDF

- in document 1:
  - term ’application’ occurs once, and in the whole collection it occurs in 2 documents:
    - tfidf (application, d1) = 1 * log(10/2) = log 5 ~ 0.7
  - term ´current´ occurs once, in the whole collection in 9 documents:
    - tfidf(current, d1) = 1 * log(10/9) ~ 0.05

13

Weighting terms: TF*IDF

- if there were some word that occurs 7 times in doc 1 and only in doc 1, the TF*IDF weight would be:
  - tfidf(doc1word, d1) = 7 * log(10/1) = 7

14

Weighting terms: normalization

- in order for the weights to fall in the [0,1] interval, the weights are often normalized (T is the set of terms):

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} (tfidf(t_s, d_j))^2}}$$

15

3. Text categorization

- problem setting
- two examples
- two major approaches
- next time: machine learning approach to text categorization

16

Text categorization

- text classification, topic classification/spotting/detection
- problem setting:
  - assume: a predefined set of categories, a set of documents
  - label each document with one (or more) categories

17

Text categorization

- let
  - D: a collection of documents
  - C = $\{c_1, \ldots, c_{|C|}\}$: a set of predefined categories
  - T = true, F = false
- the task is to approximate the unknown target function Φ’: D x C -> {T,F} by means of a function Φ : D x C -> {T,F}, such that the functions ”coincide as much as possible”
- function Φ’ : how documents should be classified
- function Φ : classifier (hypothesis, model...)

18

Example

- for instance
  - categorizing newspaper articles based on the topic area, e.g. into the 17 “IPTC” categories:
    - Arts, culture and entertainment
    - Crime, law and justice
    - Disaster and accident
    - Economy, business and finance
    - Education
    - Environmental issue
    - Health
    - ...

19

Example

- categorization can be hierarchical
  - Arts, culture and entertainment
    - archaeology
    - architecture
    - bullfighting
    - festive event (including carnival)
    - cinema
    - dance
    - fashion
    - ...

20

Example

- ”Bullfighting as we know it today, started in the village squares, and became formalised, with the building of the bullring in Ronda in the late 18th century. From that time,...”
- class:
  - Arts, culture and entertainment
  - Bullfighting
  - or both?

21

Example

- another example: filtering spam
- ”Subject: Congratulation! You are selected!
  - It’s Totally FREE! EMAIL LIST MANAGING SOFTWARE! EMAIL ADDRESSES RETRIEVER from web! GREATEST FREE STUFF!”
- two classes only: Spam and Not-spam

22

Text categorization

- two major approaches:
  - knowledge engineering -> end of 80’s
    - manually defined set of rules encoding expert knowledge on how to classify documents under the given gategories
      - If the document contains word ’wheat’, then it is about agriculture
  - machine learning, 90’s ->
    - an automatic text classifier is built by learning, from a set of preclassified documents, the characteristics of the categories

23

Text categorization

- Next lecture: machine learning approach to text categorization

24