Ling 131A
Introduction to NLP with Python

Classifiers

Marc Verhagen, Fall 2018

Today

- Final project
- Quiz 2
- Unanswered question
- Features to use
- Vector space model and TF-IDF
- Assignment 5 – word sense disambiguation

Quiz 2

- All class notes starting with WordNet
- NLTK Ch 3: 3.4 – 3.7
- NLTK Ch 5: 5.1-5.2, 5.4-5.7
- NLTK Ch 6: 6.1.1-6.1.5, 6.3-6.5
- Questions:
  - Python class, WordNet, decision trees or bayes, taggers, classifiers, vectors, evaluation, trees and grammars

Feature engineering

Temporal relation classification between events
test.xml-ei3-ei4 None e1-asp=NONE e1-cls=OCCURRENCE e1-epos=VERB e1-mod=NONE e1-pol=POS e1-stem=None e1-str=fell e1-syn=vg-s e1-tag=EVENT e1-ten=PAST e2-asp=NONE e2-cls=OCCURRENCE e2-epos=VERB e2-mod=NONE e2-pol=POS e2-stem=None e2-str=pushed e2-syn=vg-s e2-tag=EVENT e2-ten=PAST shAsp=0 shTen=0

Technology classification
2004|US6776488B2.xml|angle n doc_loc=22 doc_loc=23 doc_loc=92 last_word=angle next2_tags=,_IN next2_tags=. next2_tags=IN_NN next_n2=,_for next_n2=._^ next_n2=of_inclination next_n3=,_for_example next_n3=._^_^ next_n3=of_inclination_of plen=1 prev_Npr=inclination_of prev_V=are_at prev_V=present prev_n2=at_an prev_n2=inclination_of prev_n2=present_an prev_n3=are_at_an prev_n3=cranes_present_an prev_n3=greater_inclination_of section_loc=DESC_later section_loc=SUMMARY_later sent_loc=17-18 sent_loc=27-28 sent_loc=5-6 tag_sig=NN

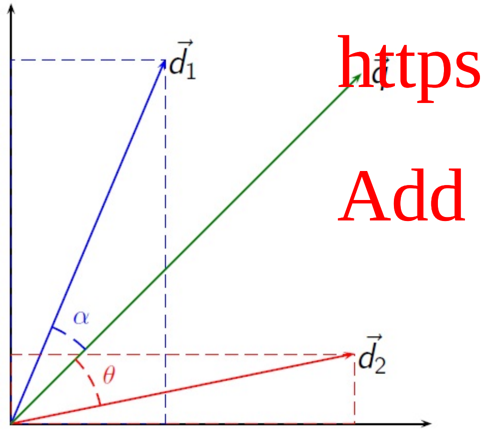
Features

- Morphological
  - Suffix: either from morphological analyzer or faking it by grabbing letters
- Word context
  - Previous_word, Next_tag
- Syntactic
  - Path_to_top, subject, predicate
  - Sometimes by using a parse, sometimes faked
- Semantic
  - WordNet sense, word class
- Metadata
  - Position in document, author

Document level

- Document level features can include
  - all kinds of meta data like author, date, publisher, topic, MESH headings, etcetera
  - words from the document, perhaps stemmed, maybe filtered with a stop list
- Vector space model is relevant here

Vector Space Model



$$cos\theta = \frac{Q \cdot D}{\parallel Q \parallel \cdot \parallel D \parallel} = \frac{\sum_{i=1}^{n} w_{q,i} w_{d,i}}{\sqrt{\sum_{i=1}^{n} w_{d,i}^2}\sqrt{\sum_{i=1}^{n} w_{q,i}^2}}$$

Vector Space Model

- Aka Term Vector Model
- Represent a text document or text passage as a vector of identifiers
- Used in information retrieval
  - Mapping a query to a set of documents
  - Both query and all documents are vectors
- Can be used for classification as well

Vectors

- Query or document regarded as a bag of terms
  - Terms can be words, lemmas, keywords, phrases

- Vector is in multi-dimensional space
  - Number of dimensions n depends on size of vocabulary

- Vector(q) = $<w_{1,q}, w_{2,q},..., w_{n,q}>$
  Vector(d) = $<w_{1,d}, w_{2,d},..., w_{n,d}>$

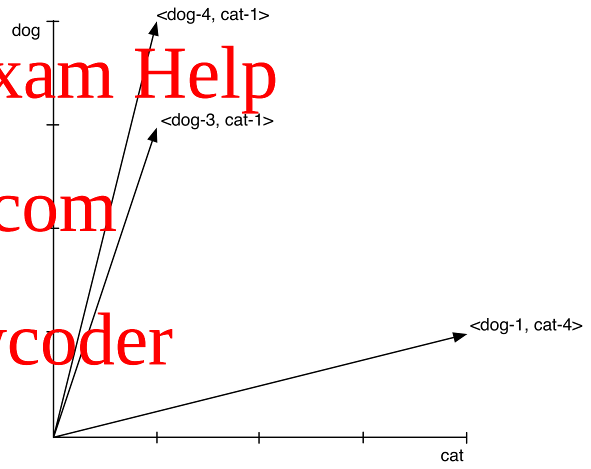
a weight is assigned to each dimension

Vectors

- Vocabulary = (dog, cat)
- Document d_1 = “dog dog dog cat”
- Weights are 0 or 1
  - – Vector(d_1) = <1,1>
- Weights are absolute frequencies
  - – Vector(d_1) = <3,1>

Similarity of vectors

- Depends on the angle between two vectors
  - The smaller the angle, the greater the similarity
  - The angle is usually calculated with the cosine measure



$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

Example calculation

cos(<dog-4 cat-1>, <dog-3 cat-1>)

A . B = Sigma(i,n) A_iB_i = 3x4 + 1x1 = 13

|A| = SQRT(Sigma(i,n) A_i^2) = SQRT(4^2 + 1^2) + SQRT(16^1) = SQRT(17) = 4.1

|B| = SQRT(Sigma(i,n) B_i^2) = SQRT(3^2 + 1^2) + SQRT(9^1) = SQRT(10) = 3.2

A.B / |A||B| = 13 / (4.1 * 3.2) = 13 / 13.04 = 0.997

cos(<dog-3 cat-1>, <dog-1 cat-4>)

A . B = Sigma(i,n) AiBi = 3x1 + 1x4 = 7

|A| = SQRT(Sigma(i,n) A_i^2) = SQRT(3^2 + 1^2) + SQRT(9 + 1) = SQRT(10) = 3.2

|B| = SQRT(Sigma(i,n) B_i^2) = SQRT(1^2 + 4^2) + SQRT(1 + 16) = SQRT(17) = 4.1

A.B / |A||B| = 7 / (3.2 * 4.1) = 7 / 13.04 = 0.537

TF-IDF

- Until now we had weights as either a binary value or a raw frequency
- Often weights are the TF-IDF score
  - Term Frequency
  - Inverse Document frequency
- Reflects how important a word is to a document in a corpus

Term Frequency

- Binary (term occurs yes/no)
- Raw count
- Adjusted for document length ($t_{f,d} = f_{t,d} / |d|$)

Inverse Document Frequency

- How much information does a word provide
- Is the term common or rare in the corpus (frequent terms count less towards the similarity scores of two documents)

- idf(t,D) = $\log_2(N/N_t) + 1$
  - N = number of documents in corpus D
  - N_t = number of documents in D with term t

TF-IDF

- Multiply the Term Frequency by the Inverse Document Frequency
- tf-idf(t,d) = $(f_{t,d} / |d|) \times (\log_2(N/N_t) + 1)$