Machine Learning
For Natural Language Processing

Abdelhak Mahmoudi
abdelhak.mahmoudi@um5.ac.ma

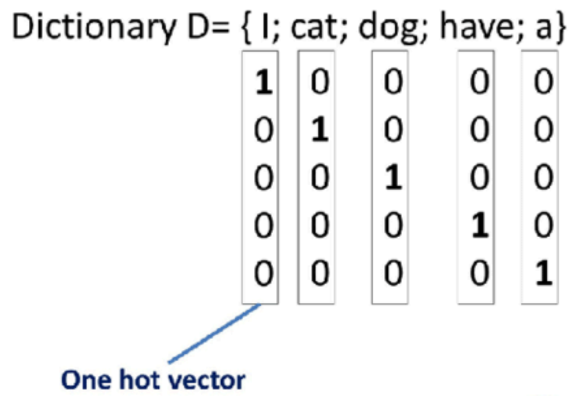
2020

Content

1. Introduction
2. Machine Learning
    1. Supervised Learning, 2. Unsupervised Learning
3. Natural Language Pr-Processing
    1. Regular Expressions, 2. Tokenization, 3. Character Encoding, 4. Part-of-Speech Tagging, 5. Chunking, 6. Stemming and Lemmatization, 7. Parsing, 8.
4. **Vector Representation of Text**
    1. **One hot vector, Word Embeddings, Tf-Idf, Word2Vec, GloVe, …**
5. Introduction to Deep Learning for NLP
    1. Sequence models, 2. BERT models
6. NLP Applications
    1. Named Entity Recognition, 9. Topic Segmentation

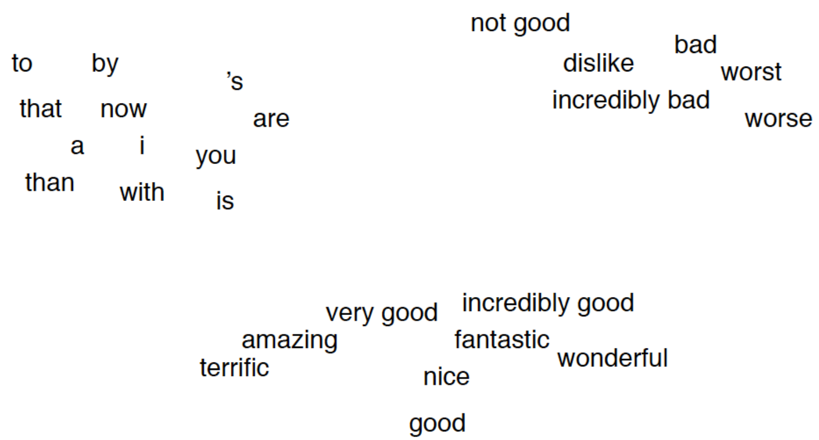
Vector Representation of Text

- Our only representation of a word is as a string of letters, or perhaps as an index in a vocabulary list (one-hot vector)
- But, we have no information about
  - Meanings (mouse, and mouse (computer))
  - Antonyms (cold/hot)
  - Synonymy (car/automobile)
  - Similarity (cat/dog)
  - Relatedness or Association (coffee and cup)
  - Connotation or Sentiment (positive or negative)
  - Etc.



Vector Representation of Text

- Solution: Vector Semantics
  - Two words that occur in very similar distributions (context) are likely to have the same meaning.
  - Define the meaning of a word w as a vector (list of numbers, a point) in N-dimensional space-> Embeddings
  - Based on counts of neighboring words
  - Tf-Idf model,
  - Word2vec model
  - Similarity measure (cosine)



- **La sémantique:** étude du sense du mot ou de la phrase.
- **La syntaxe: étude** de la forme, de la langue, de la graphie et de la grammaire du mot.

Document Representation

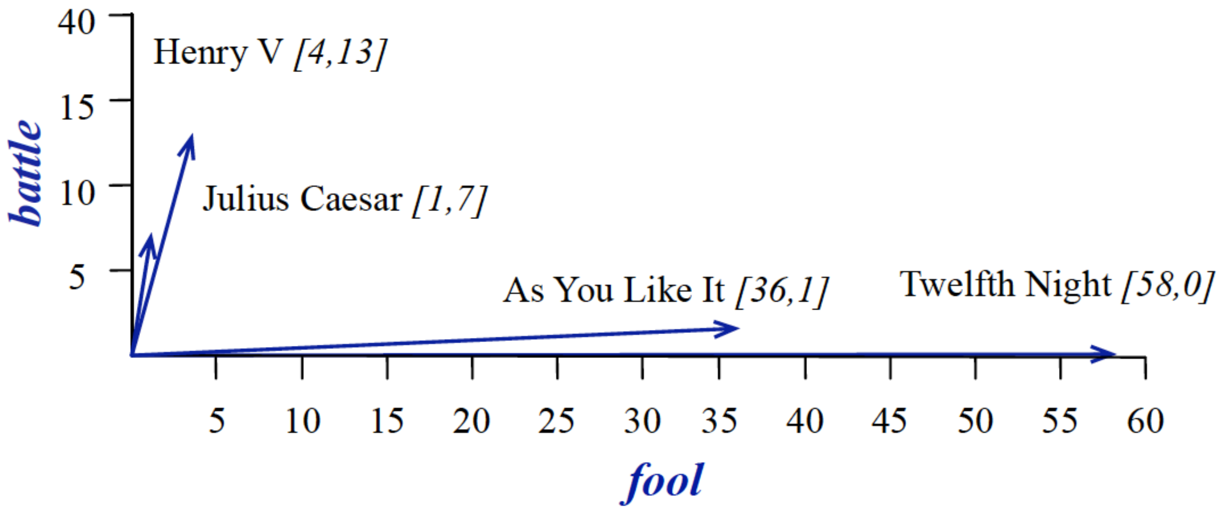
- Represent a document as a count vector in the vocabulary space
  - Term-Document Co-occurrence matrix
  - Bag-of-Words
- N: number of documents in the Corpus
  - Big Data (all the web)
- Use case
  - Information retrieval: find similar documents
- Problem
  - sparse vectors: mostly zeros

Bag-of-Words

N Documents (Corpus)

word

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3



Word Representation

- Represent a word as a count vector in the documents space.
  - Co-occurrence matrix called Term-Document matrix
- Represent a word as a count vector in the vocabulary space.
  - Co-occurrence matrix called term-term matrix, Word-word matrix or Term-matrix.
- $|V|$: vocabulary size
  - 10,000 – 50,000 words

N Documents (Corpus)

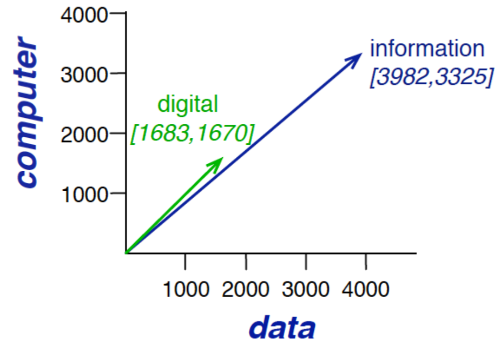
word

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Vocabulary

word

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	
strawberry	0	...	0	0	1	60	19	
digital	0	...	1670	1683	85	5	4	
information	0	...	3325	3982	378	5	13	



Term Frequency – Inverse Document Frequency

- Simple frequency isn't the best measure of association between words.
- Terms like "the, it, or they" occurs frequently !
- TF = the frequency of the term (word) t in the document d
  - TF = count(w, d)
  - $TF_{t,d}$= count(t, d)
  - $TF_{t,d} = log_{10}$(count(t, d)+1)
- IDF = is used to give a higher weight to words that occur only in a few documents
  - $IDF_t = log_{10}(\frac{N}{DF_t})$
  - DF_t = the number of documents in which term t occurs.
  - N the number of documents (in the example N = 37)

Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0

	Collection Frequency	Document Frequency
Romeo	113	1
action	113	31

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	0.074	0	0.22	0.28
good	0	0	0	0
fool	0.019	0.021	0.0036	0.0083
wit	0.049	0.044	0.018	0.022

Term Frequency – Inverse Document Frequency

- Other Alternative
  - PPMI: Positive Point wise Mutual Information
- Problem of TF-IDF
  - Ignore the order of words
  - Still Huge vector (50,000)
    - → need 10,000 weights in the network
    - May overfit
  - Still Sparse (mostly zeros)
    - May not capture synonymy
- Need
  - **Short** and **dense** vector representation -> word2vec

	Collection Frequency	Document Frequency
Romeo	113	1
action	113	31

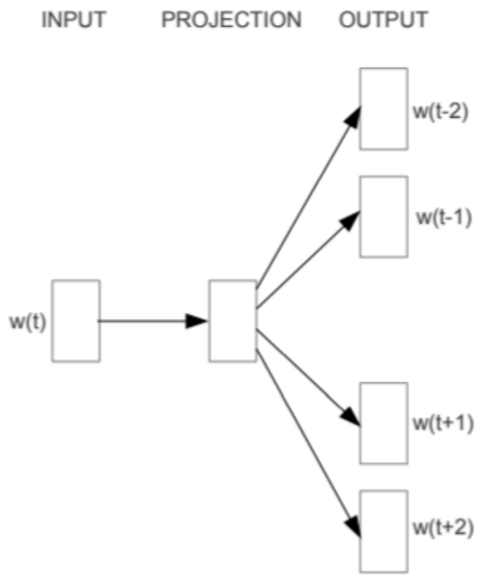
Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	0.074	0	0.22	0.28
good	0	0	0	0
fool	0.019	0.021	0.0036	0.0083
wit	0.049	0.044	0.018	0.022

Word2vec Representation

- Skip-gram with negative sampling (SGNS)



Skip-gram

Word2vec Representation

- Skip-gram with negative sampling (SGNS)
- Intuition
  - Instead of counting how often each word w1 occurs near w2, we'll instead train a classifier (a simple logistic regression) on a binary prediction task: "Is w1 likely to show up near w2?" and take the learned classifier weights as the word embeddings.

Context of w_t

$$w_{-100}, w_{-99}, \ldots\ldots \overbrace{w_{t-1}, w_{t}, w_{t+1}}^{\text{Context of } w_t}, \ldots w_{100}$$

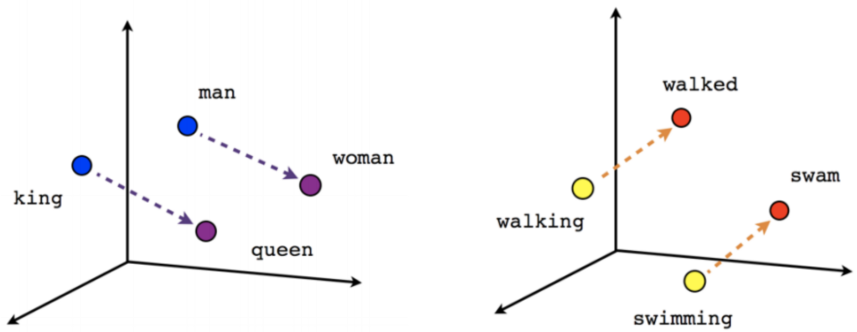
1. w_t target word
2. w_{t-1}, w_t, w_{t+1} as positive examples.
3. other words as negative samples.
4. Use logistic regression to train a classifier to distinguish those two cases.
5. Use the regression weights as the embeddings.

Word2vec Representation

- Word Embedding (Word2Vec)
  - Map words to vectors
  - Take into account word's context and position
  - Cosine Similarity

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

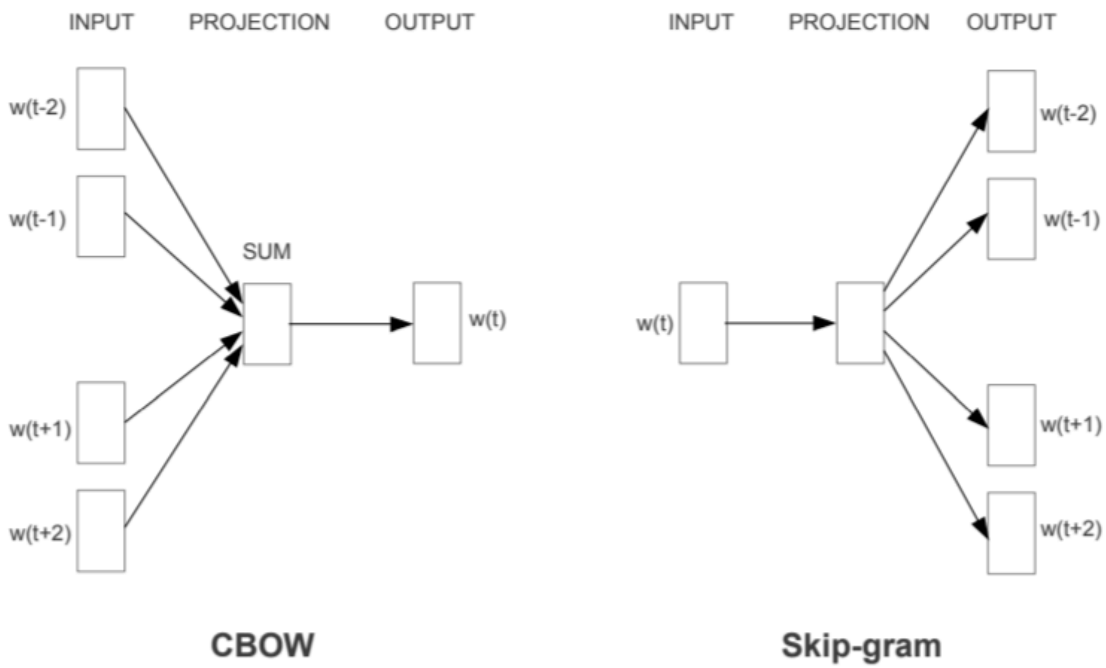
Word Embedding



Male-Female

Verb tense

Contect Bag of Word (CBoW)



GloVe Representation

- https://nlp.stanford.edu/projects/glove/

FastText Representation

- https://fasttext.cc/