Big Data Technology

Text Analytics

Paul Rad, Ph.D.

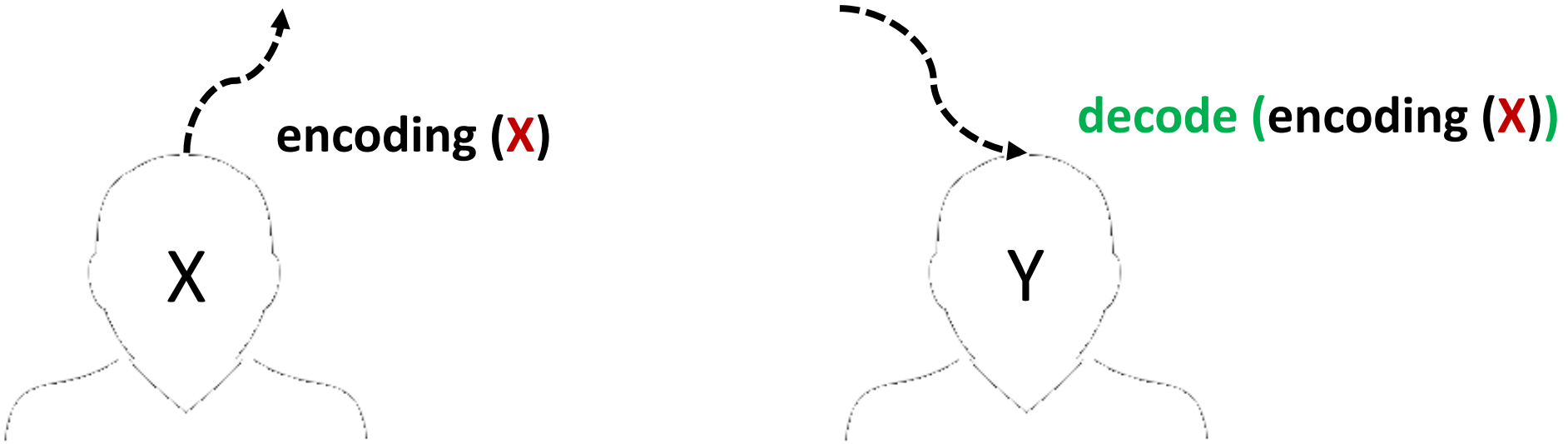
Associate Professor
Information Systems and Cyber Security, College of Business School
Electrical and Computer Engineering, College of Engineering

Outline

- What is Text
- Tokenization
- Token normalization
- Stemming and Lemmatization
- Bag of Words
- Vector representation
- word embedding
- Distance/similarity

Semantic Analysis

X: One morning I shot an elephant in my pajamas



Communication involves **recursive reasoning**: how can X choose words to maximize understanding by Y?

What is text?

Construct of text as a sequence of:

- Characters
- Words
- Phrases and names entities
- Sentences
- Paragraphs

What is a word?

Word

- Meaningful sequence of characters

-Text as a sequence of words

Sentence

A set of words that is complete in itself, typically containing a subject and predicate, conveying a statement, question, exclamation, or command.

In English we can split a sentence by spaces or punctuation.

In Japanese there are no spaces at all!

Tokenization

is a way to split text into tokens. These tokens could be **paragraphs**, **sentences**, or **individual words**.

A token is a useful unit for semantic processing

Token normalization

Stemming In linguistic morphology and information retrieval, **stemming** is the process for reducing inflected (or sometimes derived) words to their **stem, base or root form.**

- The most common algorithm for stemming English, and one that has repeatedly been shown to be empirically very effective, is *Porter's algorithm* (Porter, 1980).
- Porter's algorithm consists of 5 phases of word reductions, applied sequentially.
- The removal of the inflectional ending from words (strip off any affixes)
  - *Laughing, laugh, laughs, laughed → laugh*
- Problems
  - Can conflate semantically different words
  - *Gallery and gall may both be stemmed to gall*

PorterStemmer

cat --> cat

cats --> cat
lie --> lie

lying --> lie

run --> run

running --> run

city --> citi

cities --> citi

month --> month

monthly --> monthli

woman --> woman

women --> women

Token Normalization

Word Net Lemmatizer

- Uses the WordNet Database to lookup lemmas
- nltk.stem.WordNetLemmatizer
- Examples:
  - Feet → Foot
  - Wolves → Wolf
  - Cats → Cat

Stemming and Lemmatization

Stemming algorithms work by cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word. This indiscriminate cutting can be successful in some occasions, but not always, and that is why we affirm that this approach presents some limitations. Below we illustrate the method with examples in both English and Spanish.

Form	Suffix	Stem
studies	-es	studi
studying	-ing	study
niñas	-as	niñ
niñez	-ez	niñ

Lemmatization, on the other hand, takes into consideration the morphological analysis of the words. To do so, it is necessary to have detailed dictionaries which the algorithm can look through to link the form back to its lemma. Again, you can see how it works with the same example words.

Form	Morphological information	Lemma
studies	Third person, singular number, present tense of the verb study	study
studying	Gerund of the verb study	study
niñas	Feminine gender, plural number of the noun niño	niño
niñez	Singular number of the noun niñez	niñez

Bag of words (BoW)

Basic method for finding topics in a text

Can be a great way to determine the significant words in a text

The more frequent a word, the more important it might be

We loose word order

Example:

Text: "The cat is in the box. The cat likes the box. The box is over the cat.”

”the": 6, "box": 3, "cat": 3, "is": 2
"in": 1, "likes": 1, "over": 1

Bag of words - Text Vectorization

One possible approach:

- Each entry describes a document
- Attribute describe whether or not a term appears in the in the document Example: Term frequency table document

Another approach:

- Attributes represent the frequency in which a term appears in the document

Example

	Terms				
	Camera	Digital	Memory	Pixel	…
Document 1	1	1	0	1	
Document 2	1	1	0	0	
…	…	…	…	…	

	Terms				
	Camera	Digital	Memory	Print	…
Document 1	0.03	0.02	0	0.01	
Document 2	0	0.004	0	0.003	
…	…	…	…	…	

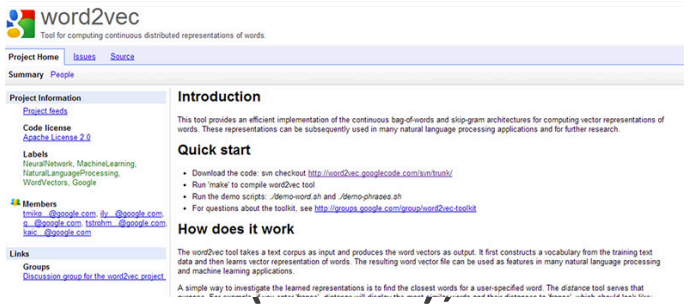
How to represent a word (word embedding)

Google Book https://code.google.com/archive/p/word2vec

- 100 billion tokens, 300 dimension, 3M words

Glove project
http://nlp.stanford.edu/projects/glove/

- Pre-trained word vectors of Wiki (6B), web cra (27B)



Vector representation

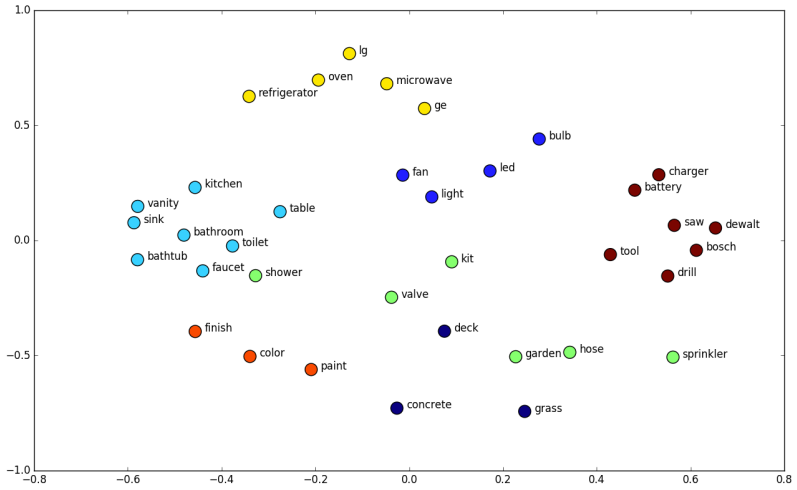
Discrete ⇒ distributed representations

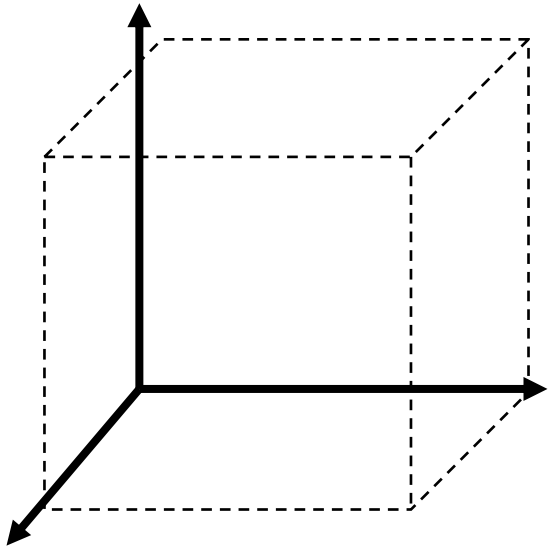
Word meanings are vector of “**basic concept**”

- What are “basic concept”?
- How to assign weights?
- How to define the similarity/distance?

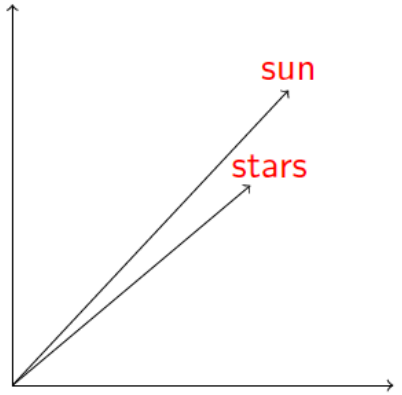
$$v_{king} = [0.8 \quad 0.9 \quad 0.1 \quad 0 \quad \dots]$$
$$v_{queen} = [0.8 \quad 0.1 \quad 0.8 \quad 0 \quad \dots]$$
$$v_{apply} = [0.1 \quad 0.2 \quad 0.1 \quad 0.8 \quad \dots]$$

royalty masculinity femininity eatable





Distance/similarity



Vector similarity measure
⇒ similarity in meaning

Cosine similarity

- $\cos(u, v) = \frac{u \cdot v}{||u|| \cdot ||v||}$

$\frac{u}{||u||}$ is a unit vector

- Word vector are normalized by length

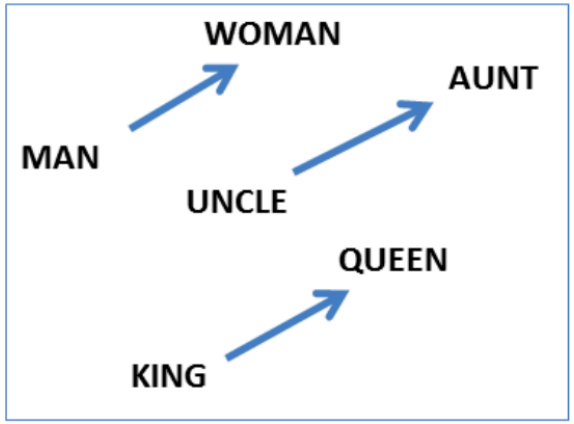
Euclidean distance $||u - v||^2$

Inner product $u \cdot v$

- Same as cosine similarity if vectors are normalized

Word analogy

$v_{man} - v_{woman} + v_{uncle} \sim v_{aunt}$



Query (a is to b as c is to d?)	Answer (d)
king : queen, man :?	woman
smart : smarter, strong :?	stronger
Tokyo : Japan, Paris :	France
Google : Larry Page, Microsoft :?	Steve Ballmer

Term Frequencies

TF-IDF weighting: give higher weight to terms that are rare

TF: term frequency (increases weight of frequent terms)

- If a term is frequent in lots of documents it does not have discriminative power

IDF: inve

For a given term w_j and document d_i

n_{ij} is the number of occurrences of w_j in document d_i

$|d_i|$ is the number of words in document d_i

n is the number of documents

n_j is the number of documents that contain w_j

$$TF_{ij} = \frac{n_{ij}}{|d_i|}$$

$$IDF_j = \log \frac{n_j}{n}$$

$$x_{ij} = TF_{ij} \cdot IDF_j$$