CSE 634

Data Mining Concepts and Techniques

Professor: Anita Wasilewska

Natural Language Processing

(From past Presentation)

References/Sources:

1. https://classroom.udacity.com/courses/ud892-preview
2. SBU CS Graduate Course: CSE 628 - Introduction to NLP (Professor Niranjan Balasubramanian)
3. Intro to Sentiment Analysis - https://www.growthaccelerationpartners.com/blog/sentiment-analysis/
4. https://en.wikipedia.org/wiki/Word2vec
5. https://www.tensorflow.org/tutorials/word2vec
6. https://www.youtube.com/watch?v=_YYQNpjvvLE&t=490s
7. http://verbs.colorado.edu/~xuen/teaching/ling5200/ppts/pos-tagging1.pdf
8. http://www.nltk.org/book/ch05.html
9. Part-of-Speech Tagging: CSE 628 Niranjan Balasubramanian
10. https://web.stanford.edu/~jurafsky/slp3/10.pdf
11. http://www.inf.ed.ac.uk/teaching/courses/inf2a/slides/2007_inf2a_L13_slides.pdf
12. http://cl.indiana.edu/~md7/13/545/slides/06-pos/06-pos.pdf
13. http://www.eng.utah.edu/~cs5340/slides/viterbi.4.pdf
14. Coursera Course on Introduction to Natural Language Processing by Prof. Dragomir Radev

Overview:

01. NLP basics
02. Language Modelling
03. POS Tagging
04. Syntactic Parsing
05. Research Paper

Introduction to NLP

Introduction

Natural-language processing (NLP)

is an area of computer science and artificial intelligence concerned

with the **interactions** between computers and human (natural) languages

In particular, is concerned with how to program computers to fruitfully

process large amounts of natural language data

Challenges in NPL
frequently involve speech recognition, natural-language understanding,
and natural-language generation

NLP is **characterized** as a hard problem in computer science as
human language is rarely precise or plainly spoken

Introduction

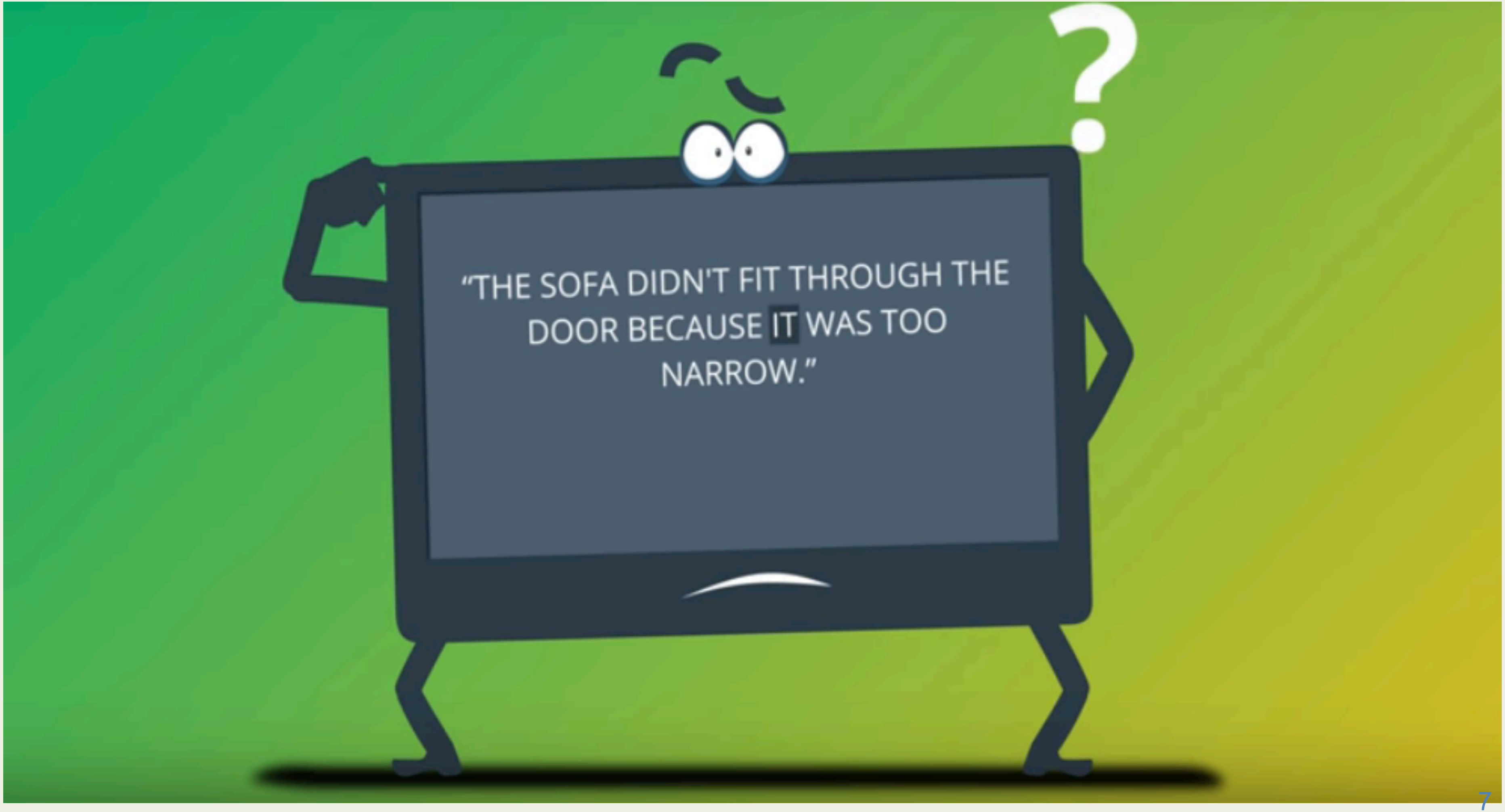
What is Natural Language Processing?

It is the field of computer science and computational linguistics
But let's take a look at a few interesting challenges

Understand semantics - --- **apply** your knowledge of the physical world

Context is everything in NLP

Here are three examples



Ref: https://classroom.udacity.com/courses/ud892-preview

Ref: https://classroom.udacity.com/courses/ud892-preview

Introduction

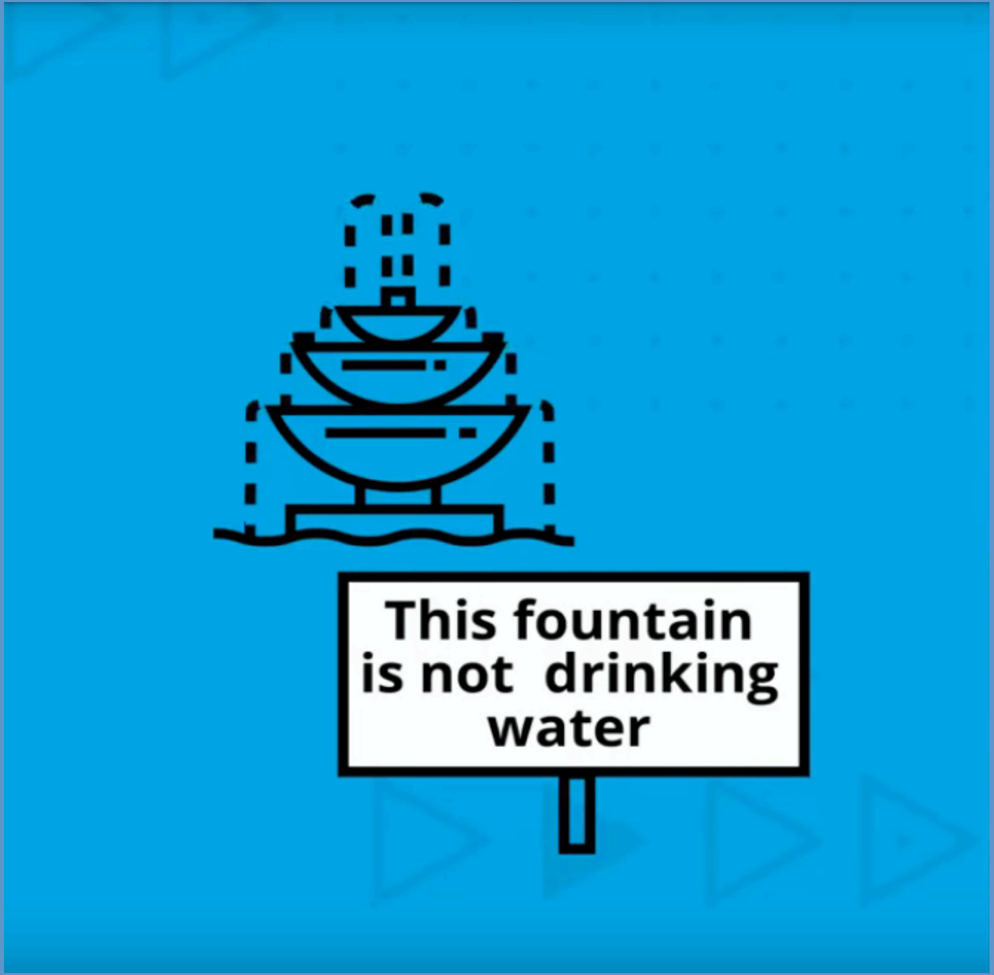
Human Understanding:

The sofa didn't fit through the door because it was too narrow.

The sofa didn't fit through the door because it was too wide.

Why do you think we are able to answer this but computer wasn’t?

- Watson demo …. https://natural-language-understanding-demo.ng.bluemix.net/



How would you interpret this one?

Ref: https://classroom.udacity.com/courses/ud892-preview

Introduction

Human Understanding:

Fountain water is not drinkable

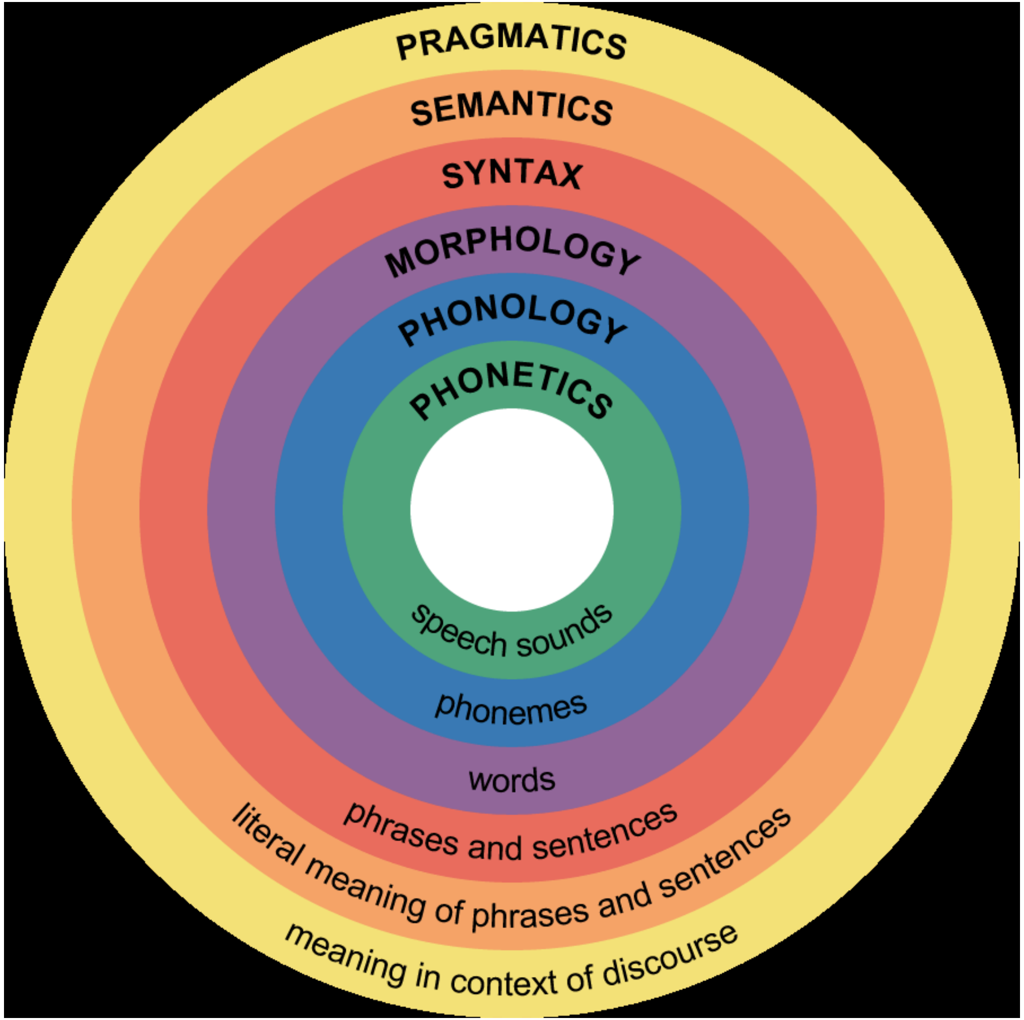
Computer Understanding:

Fountain is not engaged in drinking water

Fountain is not going to drink the water

Understand semantics - apply your knowledge of the physical world

Why do you think you were able to answer this but computer wasn't?



Challenges:

- Variability
- Ambiguity
- Meaning is context dependent
- Requires background knowledge

Ref: CSE 628 - Introduction to NLP (Professor Niranjan Balasubramanian)
Image From: Commons.wikimedia.org

Introduction

How does the communication context affect meaning?

What are the meanings of words, phrases etc.?

How do words form phrases, and phrases sentences?

How do morphemes, i.e. sub-word units, form words?

How do phonemes, i.e., sound units, form pronunciations?

How are the speech sounds generated and perceived?

Introduction

- How does the communication context affect meaning?
- What are the meanings of words, phrases etc.?
- How do words form phrases, and phrases sentences?
- How do morphemes, i.e. **sub-word** units, form words?
- How do phonemes, i.e., **sound units**, form pronunciations?
- How are the speech sounds generated and perceived?

Some NLP applications

1. Spelling and Grammar Correction/ detection (Eg. MS-Word, Grammarly etc.)

2. Machine Translation (Eg. Google Translate, Bing Translate)

3. Opinion Mining (Eg. Extract sentiment of demographic from blogs and social media)

4. Speech Recognition and Synthesis (Eg. Siri, Google assistant, Amazon Alexa)

References:
http://slideplayer.com/slide/6528602/
http://www.zydoc.com/medisapien-nlp-platform-features/

NLP Toolkits

Found around the web!

- Stanford NLP Pipeline (Java)
- spaCy (Python)
- NLTK (Python)
- Factorie and Mallet(Scala+Java)
- Apache OpenNLP (Java)
- GATE (Java)

Language Modelling

One-Hot Vectors

Machine learning algorithms work with numeric values and NLP **application** generate data in the form of words and sentences

One way to **convert** the words into numeric values is **one-hot vectors**

We take all the words that are present in the dictionary and **make vectors** such that one index represents the word and the rest all are zeros

How do we convert words to numbers?

One –Hot VECTORS

Rome Paris word V

```
Rome   = [1, 0, 0, 0, 0, 0, ..., 0]
Paris  = [0, 1, 0, 0, 0, 0, ..., 0]
Italy  = [0, 0, 1, 0, 0, 0, ..., 0]
France = [0, 0, 0, 1, 0, 0, ..., 0]
```

Problem with One-Hot Vectors

- Machine Learning algorithms using
- One-Hot Vectors are computationally expensive

- They do not consider the similarity between words

Bag –of - Words

Another approach to solve this problem is
Bag of Words

In this approach, we take a document and find out the frequencies of occurrence of words in it

And then these frequencies are fed into the machine learning algorithm

Bag-of-Words

Bag of words is a collection of all the words that are present in the document along with their frequency.

"John likes to watch movies. Mary likes movies too."

```
BoW1 = {"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1};
```

Then we use the frequencies as a feature (values of attributes) in our machine learning algorithms.

Problem with Bag-of-Words

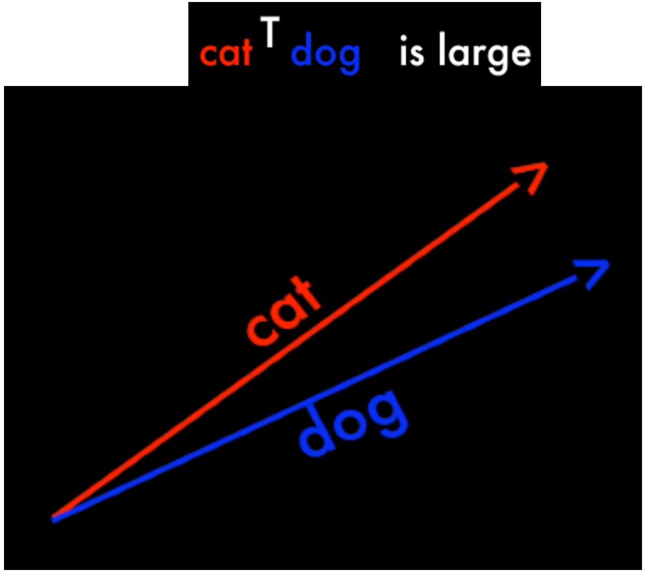
- Too simplistic
- Ignores the context of the word
- Loses the ordering of the words
- **For example:** "My name is John" is same as "Is my name John?"

Word2Vector Model

- So, to remedy these problems, engineers at Google came up with the Word2Vec model
- In this approach we represent words as vectors.
- If you two vectors are similar then their dot product is very high and as they move away from each other their dot product reduces until they are perpendicular to each other and then their dot product is zero

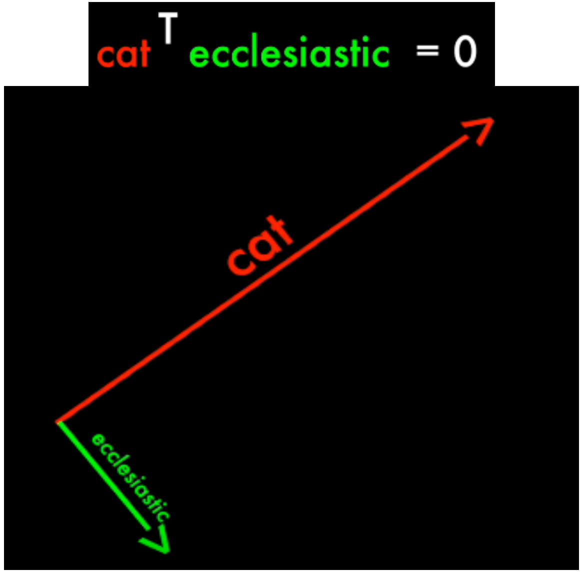
Word to Vectors

We represent every word in the form of vectors. As shown in the example below:



"CAT": 0.1, 0.5, 0.2

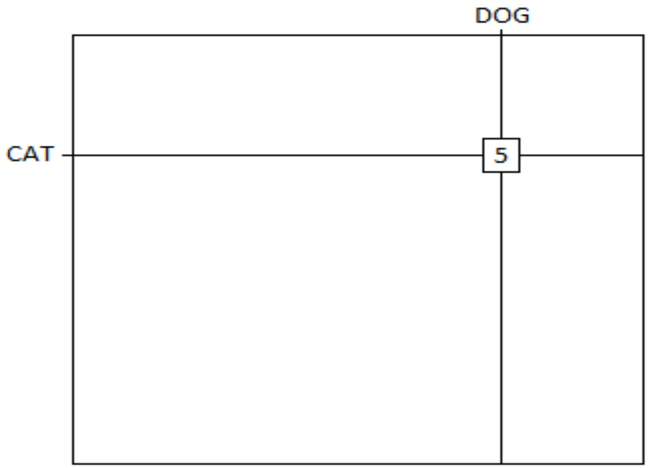
"DOG": 0.4, 0.1, 0.5



Ref: https://www.youtube.com/watch?v=_YYQNpjvvLE&t=490s

Word2Vec - Creation

Find the number of occurrences when both the words - DOG and CAT occur together



Ref: https://www.youtube.com/watch?v=_YYQNpjvvLE&t=490s

Word2Vec - Creation

"CAT" = u

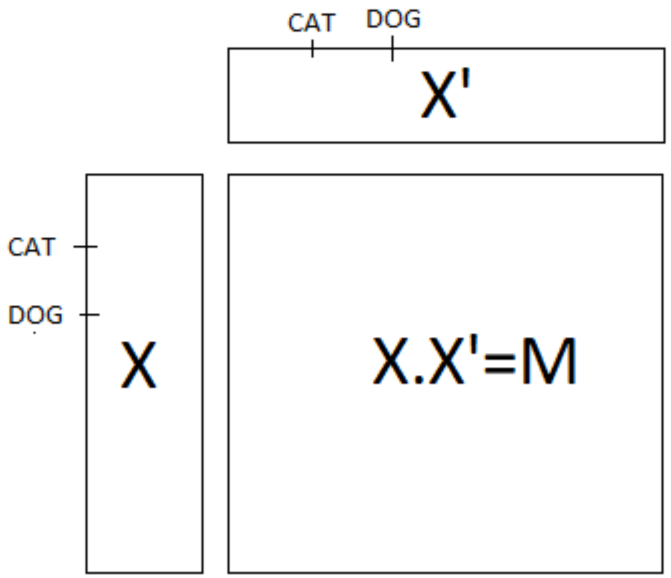
"DOG"= v

Find a value of u and v such that $u^T v$ is approximately equal to 5 which is the number of times the two words occur together.

$$u^T v = u_1.v_1+u_2.v_2+u_3.v_3 = 5$$

Ref: https://www.tensorflow.org/tutorials/word2vec

Word2Vec - Another Way



Another way to visualize this problem is through this matrix multiplication. We put our vectors for all the words in **matrix** X and take its transpose X'. When we multiply X with X' we should approximately get the matrix M

Word2Vec-Last Remarks

1. Instead of using the frequency of two words occurring together in the matrix M, we actually take the logarithm of the frequency. This helps us with words like “the”, “a”, “and” etc.

2. The biggest problem with Word2Vec is that it cannot handle new or out-of-vocabulary (OOV) words. If your model has not encountered a word before then it will have no idea how to interpret it or how to build a vector out of it. One is forced to use a random vector.

POS tagging

POS Tagging

Process of classifying words into their parts of speech and labeling them accordingly

Parts of speech are also known as word classes or lexical categories

The **collection of tags** used for a particular task is known as a tagset

Words from the same **part of sp**eech tends to behave in a similar way grammatically

References:
http://verbs.colorado.edu/~xuen/teaching/ling5200/ppts/pos-tagging1.pdf
http://www.nltk.org/book/ch05.html
Part-of-Speech Tagging: CSE 628 Niranjan Balasubramanian

Parts of Speech in English

There are several POS Tagsets

Most modern language processing on English uses the 45-tag Penn Treebank tagset (Marcus et al., 1993) as show in the table

Other tagsets:

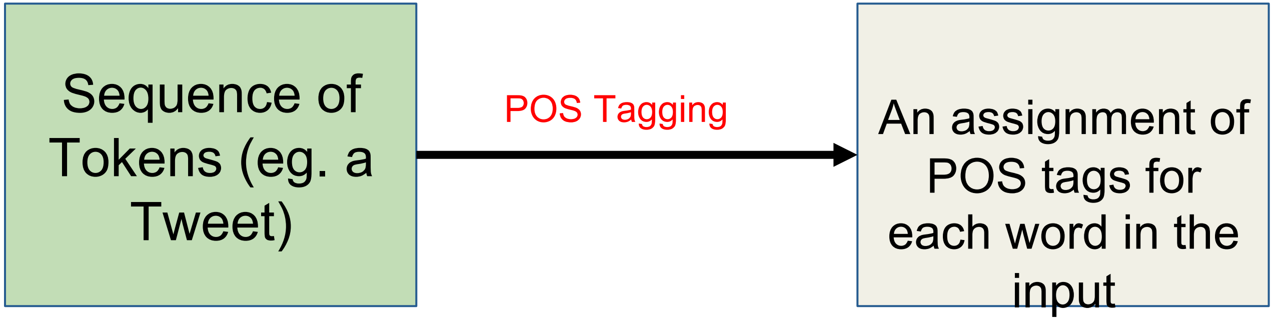
Brown corpus: 87 POS tags

C5 tagset: 61 POS tags

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two*	TO	“to”	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential ‘there’	*there*	VB	verb base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb gerund	*eating*
JJ	adjective	*yellow*	VBN	verb past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, sing.	*IBM*	$	dollar sign	$
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	“	left quote	‘ or “
POS	possessive ending	*’s*	”	right quote	’ or ”
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one’s*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

References:
https://web.stanford.edu/~jurafsky/slp3/10.pdf

POS Tagging



There/**EX** are/**VBP** 70/**CD** children/**NNS** there/**RB**

References:
Part-of-Speech Tagging: CSE 628 Niranjan Balasubramanian
https://web.stanford.edu/~jurafsky/slp3/10.pdf

Benefits of POS tagging

Succinctly characterizes the **context** of a word

Helps in recognizing similarities and differences between words

Text to speech applications: e.g. pronunciation of lead

References:
Part-of-Speech Tagging: CSE 628 Niranjan Balasubramanian
http://www.inf.ed.ac.uk/teaching/courses/inf2a/slides/2007_inf2a_L13_slides.pdf

Challenges

Same words can have different POS tags when **used** in different contexts

> **book** that flight: verb
> hand me that **book**: noun

Need to understand the meaning of the sentence **before assigning** POS tags: Difficult

Unknown/New words cannot be specified POS tags: Have to guess the tag

References:
Part-of-Speech Tagging: CSE 628 Niranjan Balasubramanian
https://web.stanford.edu/~jurafsky/slp3/10.pdf
http://cl.indiana.edu/~md7/13/545/slides/06-pos/06-pos.pdf

Rule-based POS Tagging

Depends on a dictionary that provides possible POS tags for a word,
or rules can be learned using **training data**

Ambiguity can be removed using **manually** developed rules

Example Rule:

if preceding word is ART:
    disambiguate {NOUN,VERB} as NOUN.

References:
http://www.eng.utah.edu/~cs5340/slides/viterbi.4.pdf

Statistical POS Tagging

Involves selecting most likely sequence of tags for words

We need to calculate *P(T1...Tn | w1...wn)*

Using Bayes Rule this is equal to:

$$P(T1...Tn \mid w1...wn) = \frac{P(T1...Tn) * P(w1...wn \mid T1...Tn)}{P(w1...wn)}$$

Calculating the above **probability** requires a lot of data

So, we approximate this by assuming independence based on part-of-speech tag bigrams and **lexical** generation probabilities

References:
http://www.eng.utah.edu/~cs5340/slides/viterbi.4.pdf

Parsing

Parsing Programming Languages

```
#include <stdio.h>

int main()
{
  int n, reverse = 0;

  printf("Enter a number to reverse\n");
  scanf("%d",&n);

  while (n != 0)
  {
    reverse = reverse * 10;
    reverse = reverse + n%10;
    n = n/10;
  }
  printf("Reverse of entered number is = %d\n", reverse);

  return 0;
}
```

Parsing Human Language

Rather different than computer languages

– No types for words

– No brackets around phrases

– Ambiguity

  • Words

  • Parses

– Implied information

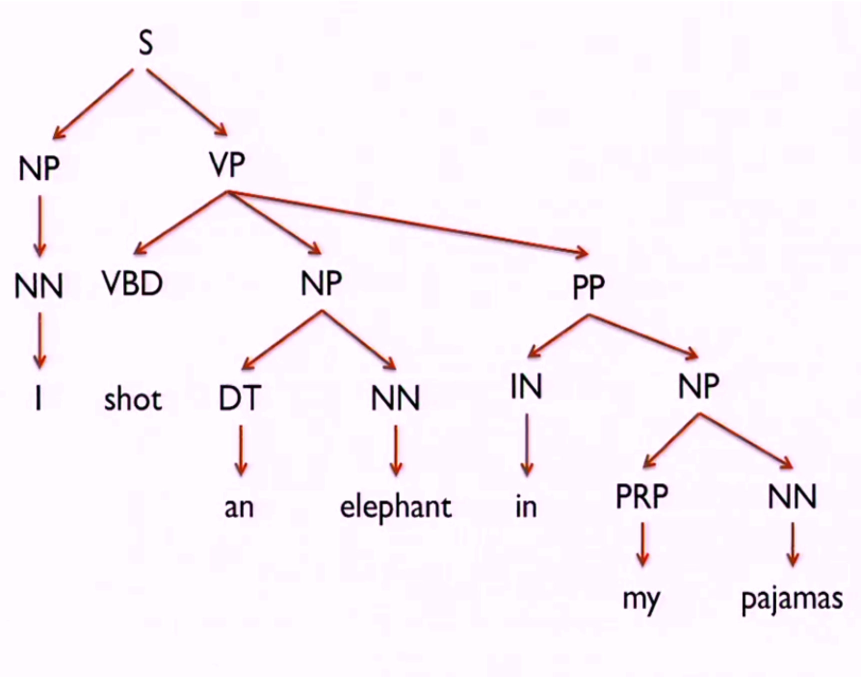
Syntactic Ambiguity

- PP attachment – I saw the man with the telescope
- Gaps – Mary likes Physics but hates Chemistry
- Coordination scope – Small boys and girls are playing
- Gerund vs. adjective – Frightening kids can cause trouble

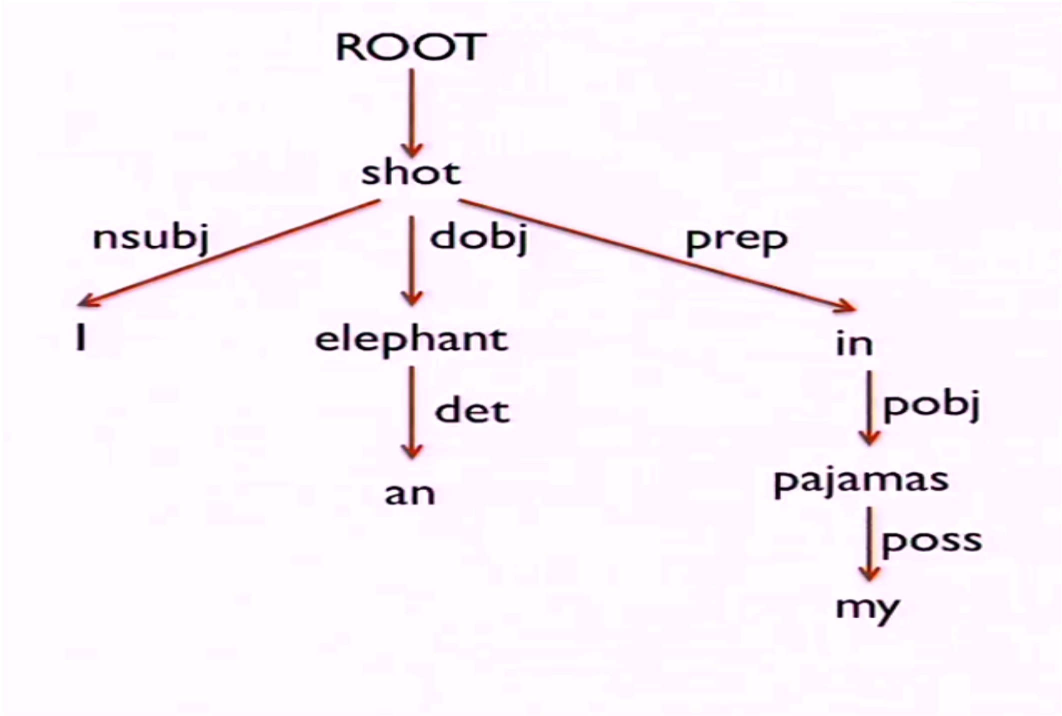
The Parsing Problem

- **Parsing** means associating tree structures to a sentence, given a **grammar** (often a Context Free Grammar)
  - There may be exactly one such tree structure
  - There may be many such structures
  - There may be none
- **Grammars** (e.g., CFG) are declarative
  - They don’t specify how the **parse tree** will be constructed

Constituency parsing



Dependency Parsing



Applications of Parsing

Constituency parsing

- Grammar checking – “I want to return this shoes”
- Machine translation – E.g., word order – SVO vs. SOV

Dependency Parsing

- Question answering – How many people in sales make $40K or more per year?

- Information extraction – Breaking Bad takes place in New Mexico.

Probabilistic CFG

Some **trees** (derivations or parses) are more **likely than** others

– Some **rules** are more frequent than others.

Argmax Pr(Tree|Sentence)

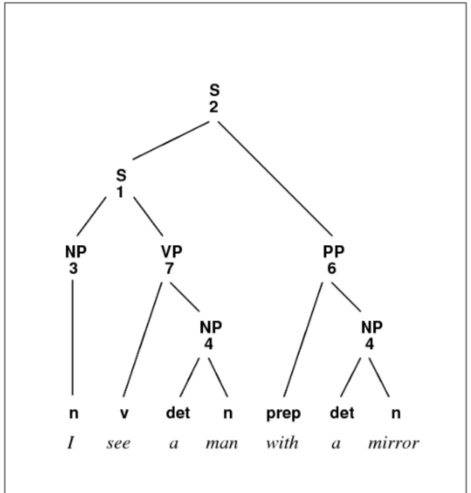


Figure 3: A parse tree

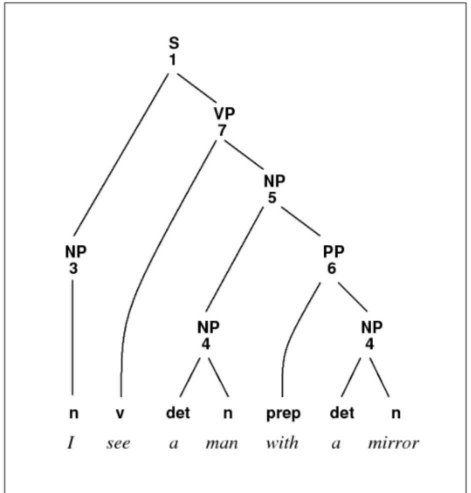


Figure 4: Another parse tree

Probabilistic CFG= CGF+Probabilities

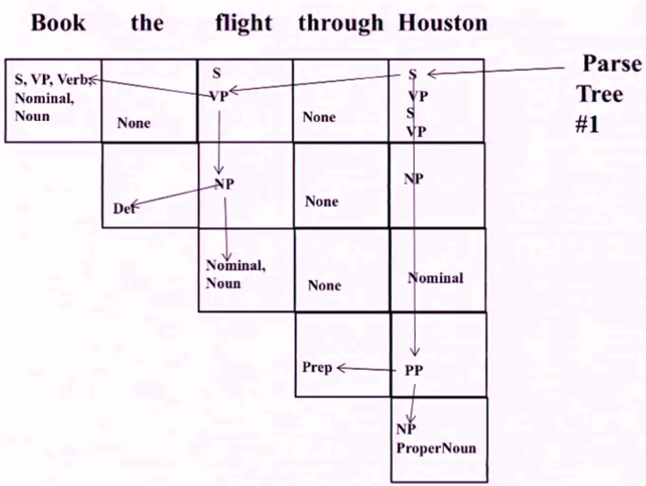
S	⇒	NP	VP	1.0
VP	⇒	Vi		0.4
VP	⇒	Vt	NP	0.4
VP	⇒	VP	PP	0.2
NP	⇒	DT	NN	0.3
NP	⇒	NP	PP	0.7
PP	⇒	P	NP	1.0

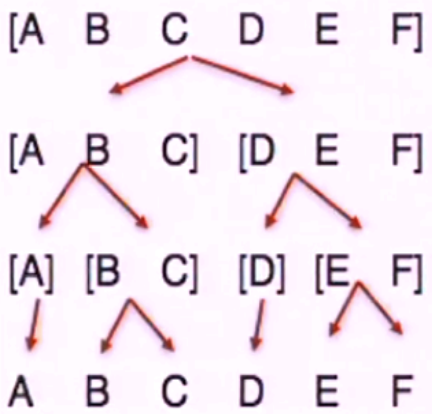
Vi	⇒	sleeps	1.0
Vt	⇒	saw	1.0
NN	⇒	man	0.7
NN	⇒	woman	0.2
NN	⇒	telescope	0.1
DT	⇒	the	1.0
IN	⇒	with	0.5
IN	⇒	in	0.5

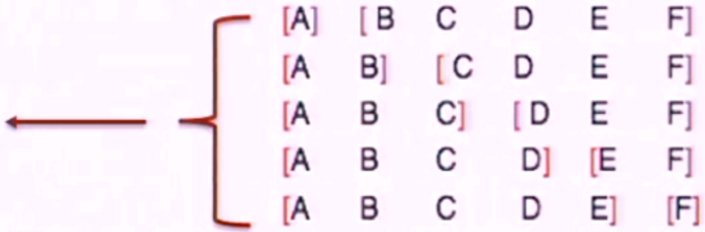
What are some ways to parse given a CFG?

- Top down parsing
- Bottom up parsing
- Dynamic Programming
  - CYK or CKY parsing [Bottom up]
  - Earley Algorithm [Top down]

CYK Parsing







The worst case complexity =θ(n³.|G|)}

Sentiment Analysis in Facebook and its application to e-learning

Sentiment Analysis

Streams of text - Customer Reviews, Social Media posts , Tweets etc

Determine how people feel about the service or product

Identify the online **mood** - positive, negative or indifferent - known as **Polarity**

Example -

"I love it" - Positive

"It is a terrible movie" - Negative

Sentiment Analysis

Accuracy is influenced by the **context** in which the words are used

Example "You must read the book" - Positive or negative?

Position of words in text is interesting to consider

Example "This book is addictive, it can be read in one sitting, but I have to admit that it is rubbish"

Presence of figures of speech - **Irony & Sarcasm**

Objective

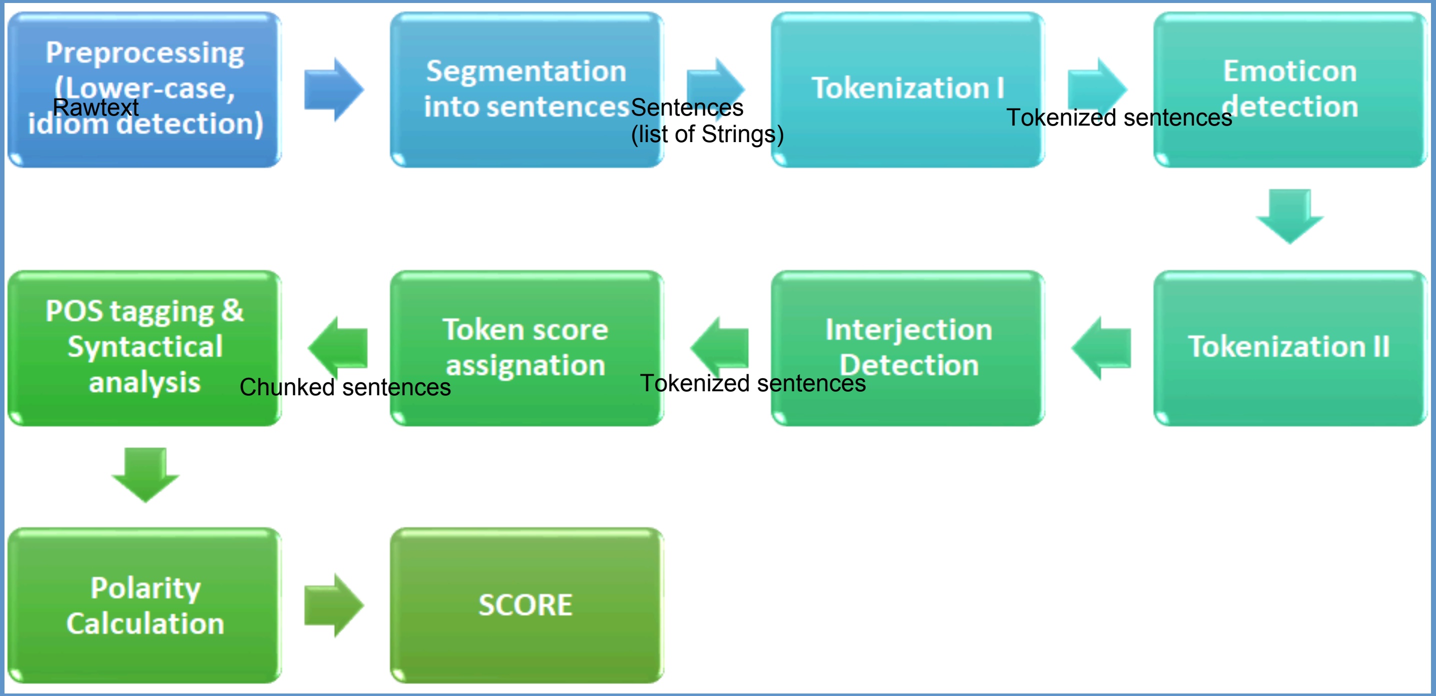
To **extract** information about the users' sentiment polarity

To **detect** significant emotional changes

How ?? - SentBuk - FB application that retrieves messages written by users and classifies them according to their polarity

Approach ?? - Lexicon based

Extract users' sentiment polarity



Extract users' sentiment polarity

Preprocessing - convert all words to lowercase

Segmentation - message divided into sentences

Tokenization I - tokens are extracted from each sentence, just spaces

Emoticon detection - classifier searches for all emoticons from text files

Tokenization II - all punctuations are considered as separators

Interjection detection - interjections are intensified by repeating letter; use regular expression to detect interjection Eg: haha vs hahahahahah

Extract users' sentiment polarity

Token Score assignation - +1 if it transmits a positive sentiment, 0 for neutral and -1 for negative sentiment

Syntactical analysis - Apply POS tagging to discriminate words that do not reflect any sentiment (Eg. articles), negations are detected (Eg. "do not like")

Polarity calculation - tokens that are susceptible to convey sentiments according to grammatical category are taken

Sum of scores divided by sum of all candidates to receive a score

Sentiment Change detection

Collect other data related to users' action that could give clues

Number of messages written (posts)

Number of comments to messages

Number of Likes made to messages, comments

Finding patterns over time - 1 day, 3-4 day, during the weekends

For example - If a user usually writes two or three messages per week and one week writes twenty messages, then this may be a sign that something different is happening to him/her

Results & Conclusion

Predicted class	Actual class				
	Positive	Neutral	Negative	Another language	Accuracy (Spanish) (%)
Positive	920	23	19	38	95.63
Neutral	74	704	65	157	83.51
Negative	89	109	760	42	79.33

Some messages **classified** as negative included irony or tease - should not be considered as **negative**

Many positive classified messages were the greetings - had high influence on the results

Changed focus on messages wrote on his/her own wall (posts)

Applications to e-learning

Gather **accurate** sentiment/opinion about the reviews for the courses and professors

Students can receive personalized advice about which educational activity to be carried out next

Motivational actions intended to encourage the **students**