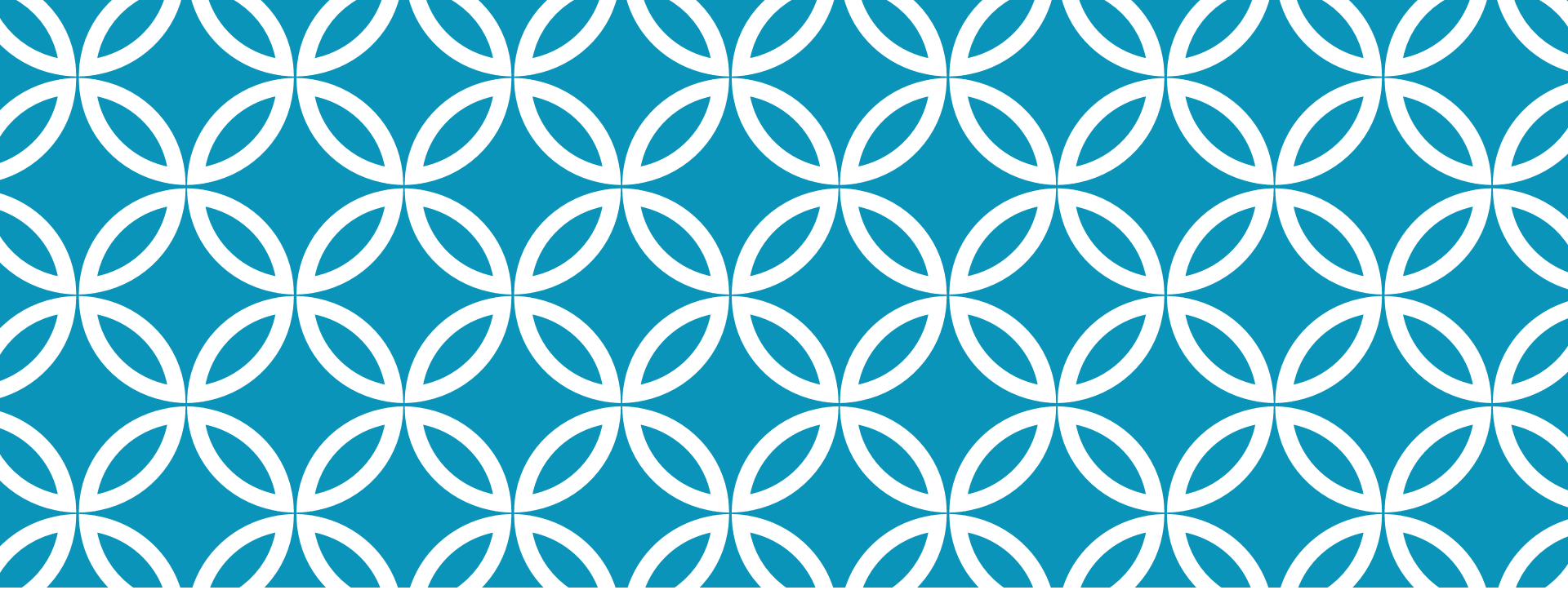


COMPUTATIONAL LINGUISTICS

by Lucas G. Freitas

COMPUTATIONAL LINGUISTICS

Linguistics

- The study of human (natural) language
- Almost 7,000 of them out there
- Syntax, semantics, pragmatics
- We learn language from minimal input in our early childhood years

Computer Science

- Artificial intelligence and machine learning
- Natural language processing
- Can you teach a computer how to speak a language?

NATURAL LANGUAGES

(1) a. Mary puts on a coat every time she leaves the house.

she = Mary: possible

b. She puts on a coat every time Mary leaves the house.

she = Mary: impossible

(2) a. Every time Mary leaves the house she puts on a coat.

she = Mary: possible

b. Every time she leaves the house Mary puts on a coat.

she = Mary: possible

Credits to Professor Maria Polinsky, Harvard Department of Linguistics

NATURAL LANGUAGES

- We don't need to have taken syntax classes to know that the second sentence is ungrammatical
- Maybe we know that a finite number of sentences are grammatically correct
- The five-headed CS50 TFs cooked the blind octopus using a DAPA mug
- Grammatical, but it doesn't seem that we have heard that sentence before
- Can computers learn languages?

CAN COMPUTERS LEARN LANGUAGES?

Native speaker

- Learns a language from exposure to it
- Has a brain with language-learning capability
- Learn from trial and error

Computer

- Can take language as input
- Then create a computer that has functions and methods that mimic each of those capabilities
- Reinforcement learning

WHILE COMPUTERS STILL CAN'T TALK…

- What are cool things we can do with language?
  - Data classification
  - Natural Language Processing
  - Machine Translation
  - So much more!

WHAT WE WILL GO OVER TODAY

- The classification problem: given the lyrics from a song, can I guess with high probability who sings it?
- The parsing problem: how to segment Chinese and other unsegmented languages?
- Syntax: how to check if a sentence is grammatical or not?
- Machine translation: how to translate something from Portuguese to English?

THE CLASSIFICATION PROBLEM

- Given some lyrics to a song, can I guess who sings it?
- We can assume that artists sing about recurrent themes, probably using recurrent words
- Think about your friends
- There is no exact answer of how to do that, but we can make some assumptions and create a model to try to accomplish that.
- We can then test the model and check its accuracy

THE CLASSIFICATION PROBLEM

- Example: people say that Lady Gaga uses the word “baby” in all of her number one songs. In fact:

LYRICS TO A SONG: “BABY”. WHO SINGS IT?

Lady Gaga

I’m on the right track, baby
I was born this way
Oh there ain’t no other way
Baby I was born this way

Katy Perry

I got the eye of the tiger the fire
Dancing through the fire
Cause I am a champion
And you’re gonna hear me roar
Louder louder than a lion

THE CLASSIFICATION PROBLEM

- Seems that Gaga is probably the singer
- How can we write that more formally?
- P(Gaga | baby)
- Bayes theorem: P(Gaga | baby) = P(baby | Gaga) x P(Gaga) / P(baby)
- P(Gaga) is called the prior probability of the class
- P(Gaga | baby) is called the posterior probability of the class

THE CLASSIFICATION PROBLEM

Lady Gaga

P(Gaga | baby) = P(baby | Gaga) x P(Gaga) / P(baby)

P(Gaga | baby) = (2 / 24) x (1 / 2) / (2 / 53)

Katy Perry

P(Katy| baby) = P(baby | Katy) x P(Katy) / P(baby)

P(Katy| baby) = (0 / 29) x (1 / 2) / (2 / 53) = 0

Gaga wins!

THE CLASSIFICATION PROBLEM

- What if we have multiple words? "baby I am on fire"
- Assume independency
- Look for candidate with the highest posterior probability. For one word:

$$\arg\max \Pr(class \mid word) = \arg\max \frac{\Pr(word \mid class)\Pr(class)}{\Pr(word)} = \arg\max \Pr(word \mid class)\Pr(class)$$

- For multiple words:

$$\text{argmax} \Pr(c_i) \prod_{j=i}^{m} \Pr(x_j \mid c_i)$$

THE CLASSIFICATION PROBLEM

- $Pr(x_j|c_j) = \#(x_j \text{ in } c_i) / \#(\text{words in } c_i)$
- Laplace Smoothing: $Pr(x_j|c_j) = (1 + \#(x_j \text{ in } c_i)) / \#(\text{words in } c_i)$

CODING TIME

THE SEGMENTATION PROBLEM

Unsegmented

椅子の上

上田さん

This is how we get data from a lot of East-Asian languages

Segmented

椅子　の　上

上田　さん

Essential for machine translation or natural language processing

LANGUAGE MODEL

- Model that describes the probability of each and every sentence in a language existing.
- Depends both on word frequency and word order
- Ex: much more common to see the word “square” than “trapezoid” in English.
- Ex: much more common to see “the cat is inside the box” than “the box inside is the cat”

N-GRAM MODELS

- The probability of a word appearing depends on the word frequency and the words around it.
- Example: "the CS50 TF"

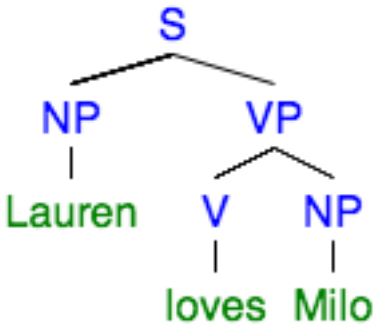
$$\Pr(w) = \Pr(w_1)\Pr(w_2 \mid w_1)P(w_3 \mid w_1 w_2)$$

TANGO SEGMENTATION

- Threshold And maximum for N-Grams that Overlap
- **椅子の上**
- Check if segmenting on each position gives us a higher n-gram probability

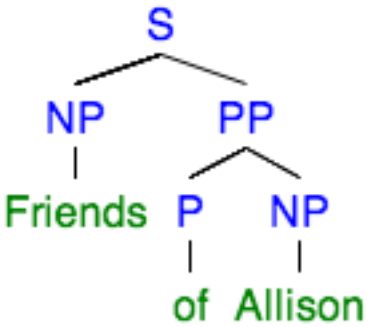
SYNTAX

- Sentence constituents
- Lexical types (noun, adjective, etc)
- Phrasal types (noun phrase, verb phrase, etc)
- Example: Lauren loves Milo
- Nouns: Lauren, Milo
- Verb: loves



WHAT CAN WE DO WITH THAT?

- Teach computer how to do syntactic classification
- Explain relations using syntactic relations
- Ex: friends of Allison
- NP_1 PP[P NP_2], P = of means that NP_2 "owns" NP_1
- This is the essence of Facebook's Graph Search!



MACHINE TRANSLATION

- Also mostly statistical
- Naïve implementation: first, translate word-by-word
- Example: o gato branco fugiu → the cat white ran away
- Now, figure out the ideal position of each word (language model)
- the white cat ran away
- Naïve method: calculate the probability of sentences for each possible word positions, choose the one with the highest

HOW CAN I GET STARTED?

- Choose a project that involves language
- Think of what kind of model you can use, and do some internet research on models that are already proven to work well for those cases
- Talk to me about your ideas and I'll be happy to advise you (me@lfreitas.com)
- Get involved with computational linguistics and you'll be happy forever! ☺