NLP For Rubyists

Aja Hammerly
@the_thagomizer
Aja Hammerly
http://github.com/thagomizer
@the_thagomizer
http://www.thagomizer.com
Lawyer Cat Says:
Any code is copyright Google and licensed Apache V2
NLP?!?!
Natural Language Processing
Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages and, in particular, concerned with programming computers to fruitfully process large natural language corpora.
- Wikipedia
Teaching computers to understand (and ideally respond to) human languages.
- Aja
Why Should I Care?
It Is Already Here
Better User Experience
Accessibility
Improved Understanding
Assist Us
Hard
Why Is This Hard?
English Is Horrible
Seal
Their
There
They're
There
They're
Their
Love
English is Horrible
All Human Languages Are Horrible
No Computational Grammar
I'm Starving
You look freezing.
Unique
Computers Suck at Sarcasm
Sure, I'd *love* to.
Why Is This Hard?
History
Leibniz & Descartes
The Turing Test
Chat Bots
Show Me The Codes
Impractical
To the

AT SCALE
sentiment analysis (noun)
the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc., is positive, negative, or neutral.
Google Cloud
Natural Language API
gem install google-cloud language
require "google/cloud/language"
language = Google::Cloud::Language.new

def analyze tweet
  document = language.document(tweet)
  sentiment = document.sentiment
  sentiment.score
end
Hand Wave
Demo
#railsconf
Sentence Diagramming
The cat ate fish.
The cat ate the fish.
Eww! Grammar.
Side Quest!!!!
Grammar 101
Parts of Speech
Verb
Action
State of Being
Noun
Person
Place
Thing
Idea
Adjective
Attributes
Compare
Article
Determiner
Parts of a Sentence
Root
Subject
Direct Object
The cat eats fish.
The cat eats fish.
Side Quest Complete
Sentence Diagramming
Subject | Verb | Direct Object

Other | Other
Syntax
require "google/cloud/language"
l = Google::Cloud::Language.new

d = l.document("The cat ate fish.")

s = d.syntax
puts s.tokens
<Google::Cloud::Language::Annotation::Token:0x007fb86d0909c0
  @head_token_index=2,
  @label=:NSUBJ,
  @lemma="cat",
  @part_of_speech=
    #<Google::Cloud::Language::Annotation::PartOfSpeech:0x007fb86d090a10
      @aspect=:ASPECT_UNKNOWN,
      @case=:CASE_UNKNOWN,
      @form=:FORM_UNKNOWN,
      @gender=:GENDER_UNKNOWN,
      @mood=:MOOD_UNKNOWN,
      @number=:SINGULAR,
      @person=:PERSON_UNKNOWN,
      @proper=:PROPER_UNKNOWN,
      @reciprocity=:RECIPROCITY_UNKNOWN,
      @tag=:NOUN,
      @tense=:TENSE_UNKNOWN,
      @voice=:VOICE_UNKNOWN>,
  @text_span=
    #<Google::Cloud::Language::Annotation::TextSpan:0x007fb86d090a60
      @offset=4,
      @text="cat">>
cat
<Google::Cloud::Language::Annotation::Token:0x007fb86d0909c0
@head_token_index=2,
@label=:NSUBJ,
@lemma="cat",
@part_of_speech=
  #<Google::Cloud::Language::Annotation::PartOfSpeech:0x007fb86d090a10
    @aspect=:ASPECT_UNKNOWN,
    @case=:CASE_UNKNOWN,
    @form=:FORM_UNKNOWN,
    @gender=:GENDER_UNKNOWN,
    @mood=:MOOD_UNKNOWN,
    @number=:SINGULAR,
    @person=:PERSON_UNKNOWN,
    @proper=:PROPER_UNKNOWN,
    @reciprocity=:RECIPROCITY_UNKNOWN,
    @tag=:NOUN,
    @tense=:TENSE_UNKNOWN,
    @voice=:VOICE_UNKNOWN>,
@text_span=
  #<Google::Cloud::Language::Annotation::TextSpan:0x007fb86d090a60
    @offset=4,
    @text="cat">>
<Google::Cloud::Language::Annotation::Token:0x007fb86d0909c0
@head_token_index=2,
@label=:NSUBJ,
@lemma="cat",
@part_of_speech=
  #<Google::Cloud::Language::Annotation::PartOfSpeech:0x007fb86d090a10
    @aspect=:ASPECT_UNKNOWN,
    @case=:CASE_UNKNOWN,
    @form=:FORM_UNKNOWN,
    @gender=:GENDER_UNKNOWN,
    @mood=:MOOD_UNKNOWN,
    @number=:SINGULAR,
    @person=:PERSON_UNKNOWN,
    @proper=:PROPER_UNKNOWN,
    @reciprocity=:RECIROCITY_UNKNOWN,
    @tag=:NOUN,
    @tense=:TENSE_UNKNOWN,
    @voice=:VOICE_UNKNOWN>,
@text_span=
  #<Google::Cloud::Language::Annotation::TextSpan:0x007fb86d090a60
    @offset=4,
    @text="cat">>
require "google/cloud/language"

language = Google::Cloud::Language.new

syn = language.document("The cat ate fish.").syntax

subj = syn.tokens.find { |t| t.label == :NSUBJ }.text_span.text
verb = syn.tokens.find { |t| t.label == :ROOT }.text_span.text

puts "#{subj} | #{verb}"
puts "----------------"
puts "#{" " * subj.size} |"
require "google/cloud/language"

l = Google::Cloud::Language.new

syn = l.document("The cat ate fish.").syntax

subj = syn.tokens.find { |t| t.label == :NSUBJ }.text_span.text
verb = syn.tokens.find { |t| t.label == :ROOT }.text_span.text

puts "#{subj} | #{verb}"
puts "----------------"
puts "#{" " * subj.size} |"
require "google/cloud/language"

l = Google::Cloud::Language.new

syn = l.document("The cat ate fish.").syntax

subj = syn.tokens.find { |t| t.label == :NSUBJ }.text_span.text
verb = syn.tokens.find { |t| t.label == :ROOT }.text_span.text

puts "#{subj} | #{verb}"
puts "----------------"
puts "#{" " * subj.size} |"
require "google/cloud/language"

l = Google::Cloud::Language.new

syn = l.document("The cat ate fish.").syntax

subj = syn.tokens.find { |t| t.label == :NSUBJ }.text_span.text
verb = syn.tokens.find { |t| t.label == :ROOT }.text_span.text

puts "#{subj} | #{verb}"
puts "----------------"
puts "#{" " * subj.size} |"
require "google/cloud/language"

l = Google::Cloud::Language.new

syn = l.document("The cat ate fish.").syntax

subj = syn.tokens.find { |t| t.label == :NSUBJ }.text_span.text
verb = syn.tokens.find { |t| t.label == :ROOT }.text_span.text

puts "#{subj} | #{verb}"
puts "----------------"
puts "#{" " * subj.size} |"
cat | ate

-------------
subj = syn.tokens.find { |t| t.label == :NSUBJ }.text_span.text
verb = syn.tokens.find { |t| t.label == :ROOT }.text_span.text
dobj = syn.tokens.find { |t| t.label == :DOBJ }.text_span.text

puts "#{subj} | #{verb} | #{dobj}"
puts "#{"-" * (subj.size + verb.size + dobj.size + 7)}"
puts "#{" " * subj.size} |"
cat | ate | fish
_______________
|
The cat eats fish.
@head_token_index=1,
@label=:DET,
@lemma="The",
@part_of_speech=

@text_span=

[The, cat, eats, fish, .]
tokens.each do |t|
  if tokens[t.head_token_index] == subj
    print t.text_span.text
  end
end
cat | ate | fish
-----------------
The |
The cat at the fish with a side of milk.
graph
node(id, label)
edge(to, from)
require "google/cloud/language"
require "graph"

l = Google::Cloud::Language.new
tokens = l.document("The cat ate fish.").syntax.tokens

digraph do
  tokens.each_with_index do |t, i|
    node(i, t.text_span.text)
    unless t.head_token_index == i
      edge(i, t.head_token_index)
    end
  end

  save "sentence4", "png"
end
require "google/cloud/language"
require "graph"

l = Google::Cloud::Language.new
tokens = l.document("The cat ate fish.").syntax.tokens

digraph do
  tokens.each_with_index do |t, i|
    node(i, t.text_span.text)
    unless t.head_token_index == i
      edge(i, t.head_token_index)
    end
  end

  save "sentence4", "png"
end
require "google/cloud/language"
require "graph"

l = Google::Cloud::Language.new
tokens = l.document("The cat ate fish.").syntax.tokens

digraph do
  tokens.each_with_index do |t, i|
    node(i, t.text_span.text)
    unless t.head_token_index == i
      edge(i, t.head_token_index)
    end
  end
end

save "sentence4", "png"
require "google/cloud/language"
require "graph"

l = Google::Cloud::Language::new
tokens = l.document("The cat ate fish.").syntax.tokens

digraph do
  tokens.each_with_index do |t, i|
    node(i, t.text_span.text)
    unless t.head_token_index == i
      edge(i, t.head_token_index)
    end
  end

  save "sentence4", "png"
end
require "google/cloud/language"
require "graph"

l = Google::Cloud::Language.new
tokens = l.document("The cat ate fish.").syntax.tokens

digraph do
  tokens.each_with_index do |t, i|
    node(i, t.text_span.text)
    unless t.head_token_index == i
      edge(i, t.head_token_index)
    end
  end

  save "sentence4", "png"
end
The cat ate fish.
The cat ate the fish on the side of a milk.
Silly Examples
Practical
Free
Google at RailsConf

- Talks:
  - Google Cloud <3 Ruby (watch on ConFreaks)
  - What’s My App *Really* Doing in Production
    - Thursday @ 3:30 in Rm 156
- Raffle for a Google Home:
Thank You
Questions?