The background of the image features the silhouettes of a man and a woman in profile, facing each other as if in conversation. The man is on the left, and the woman is on the right. The silhouettes are dark against a light, slightly textured background. Two text boxes are overlaid on the image: an orange-bordered box on the left containing a quote by Confucius, and a purple-bordered box on the right containing a quote by Lawrence Gobright. The text in both boxes is in a bold, black, sans-serif font.

**“Without
knowing the
force of words,
it is impossible
to know more.”**

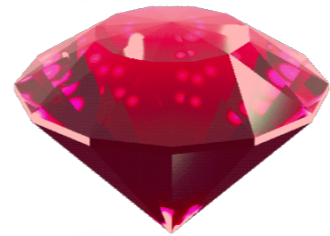
–Confucius

**“My
dispatches are
merely dry
matter of fact
and detail.”**

*–Lawrence
Gobright,
Washington AP
Chief (1856)*

@rabbigreenberg

What's LOVE Got To Do With It?



Ruby and Sentiment Analysis

Ben Greenberg

@rabbigreenberg

Who is talking right now? 🤔

(i.e. Hi! 🙋)

- I ❤️ Ruby
- Co-editor of Torah & Tech
- Former rabbi & community organizer
- 🌍 San Diego → NYC → Boston → Denver → NYC → Tel Aviv 🌍
- Developer Advocate @ Nexmo



@rabbigreenberg

Our Roadmap

- A. Why does this matter?
- B. What is sentiment analysis?
- C. What can I build with it?
- D. Let's build it!
- E. Test Run



@rabbigreenberg

Why does this
matter?

**Language is not only
a passive vehicle for information**

“Last night I had
dinner with...
someone, it
was delicious”

”אתמול בערב
אכלתי ארוחת ערב
עם...
מישהי, היא
טעיימה”

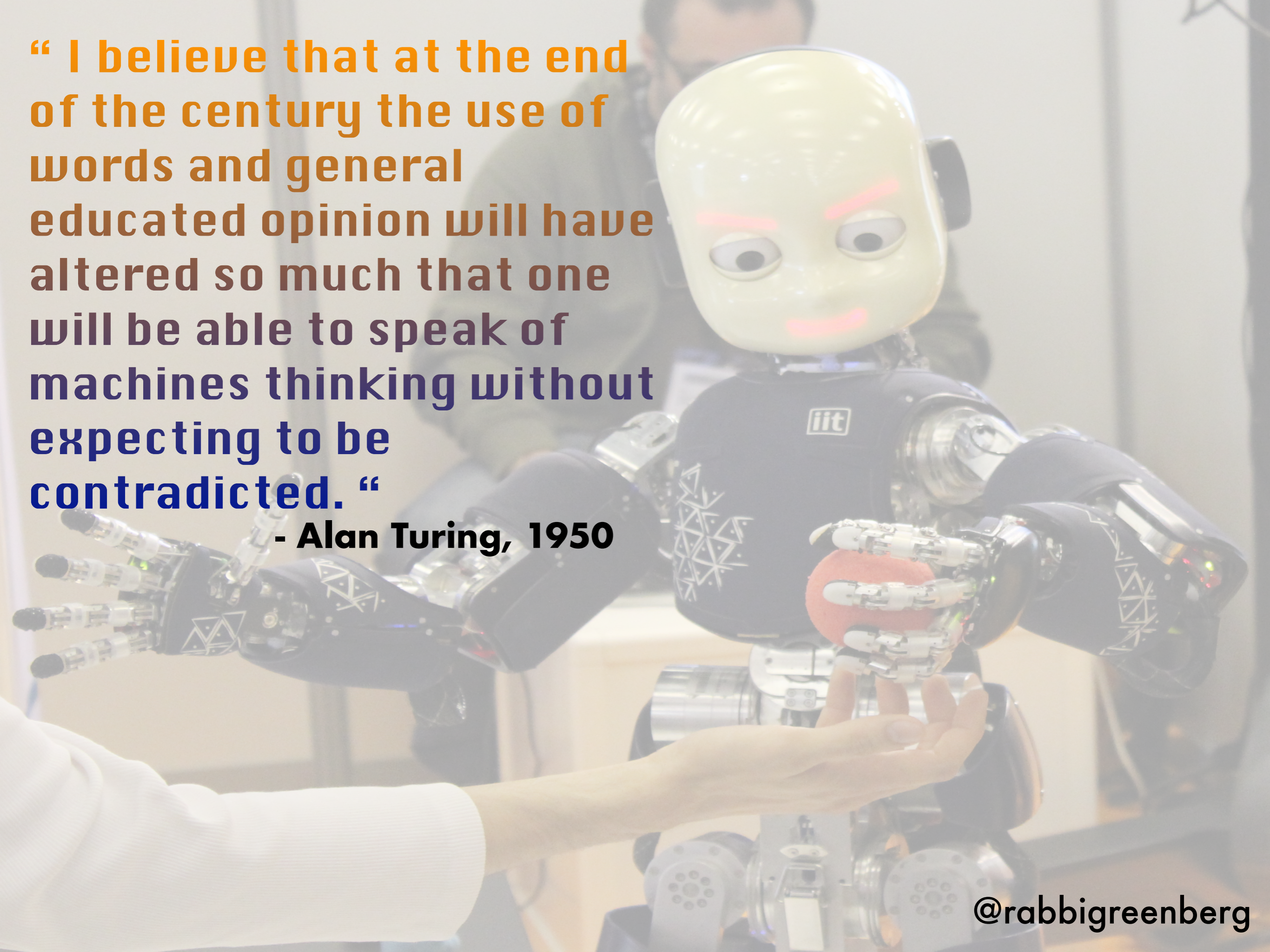


**Language is a
conduit constructor**

What is sentiment analysis?

“ I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted. “

- Alan Turing, 1950



“Last night I had dinner with someone, it was delicious”

sentiment=>{"document"=>{"score"=>0.970908, "label"=>"positive"}}

semantic_roles=> [{"subject"=>{"text"=>"I"}, "sentence"=>"Last night I had dinner with someone, and it was delicious.", "object"=>{"text"=>"dinner"}, "action"=>{"verb"=>{"text"=>"have", "tense"=>"past"}, "text"=>"had", "normalized"=>"have"}}],

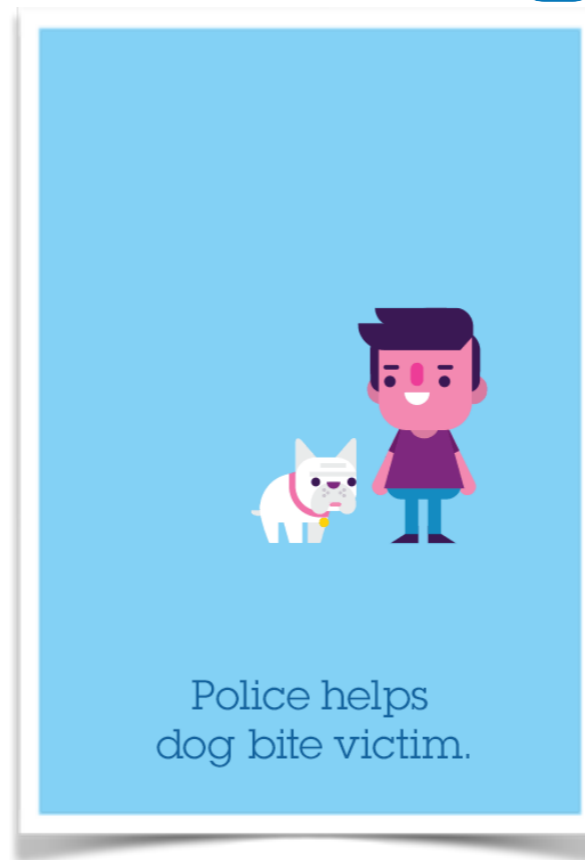
keywords=> [{"text"=>"Last night", "relevance"=>0.998604, "count"=>1}, {"text"=>"dinner", "relevance"=>0.672861, "count"=>1}],

emotion=>{"sadness"=>0.049938, "joy"=>0.857958, "fear"=>0.041413, "disgust"=>0.034346, "anger"=>0.025449}},

categories=> [{"score"=>0.658246, "label"=>"/food and drink/food", "explanation"=>{"relevant_text"=> [{"text"=>"dinner"}]}], {"score"=>0.65732, "label"=>"/food and drink/desserts and baking"}]

Natural Language Processing

- Formal Grammar
- Inflectional Endings
- Part-of-speech (POS) tagging
- Parsing sentences
- Sentence boundaries
- Root form of words
- Segment continuous text into separate words (prevalent in many languages)
- Relationships of named entities to each other (who to whom, what to what, etc.)
- Topic segmentation
- Word meaning probability



I'M OVERLY
EMOTIONAL

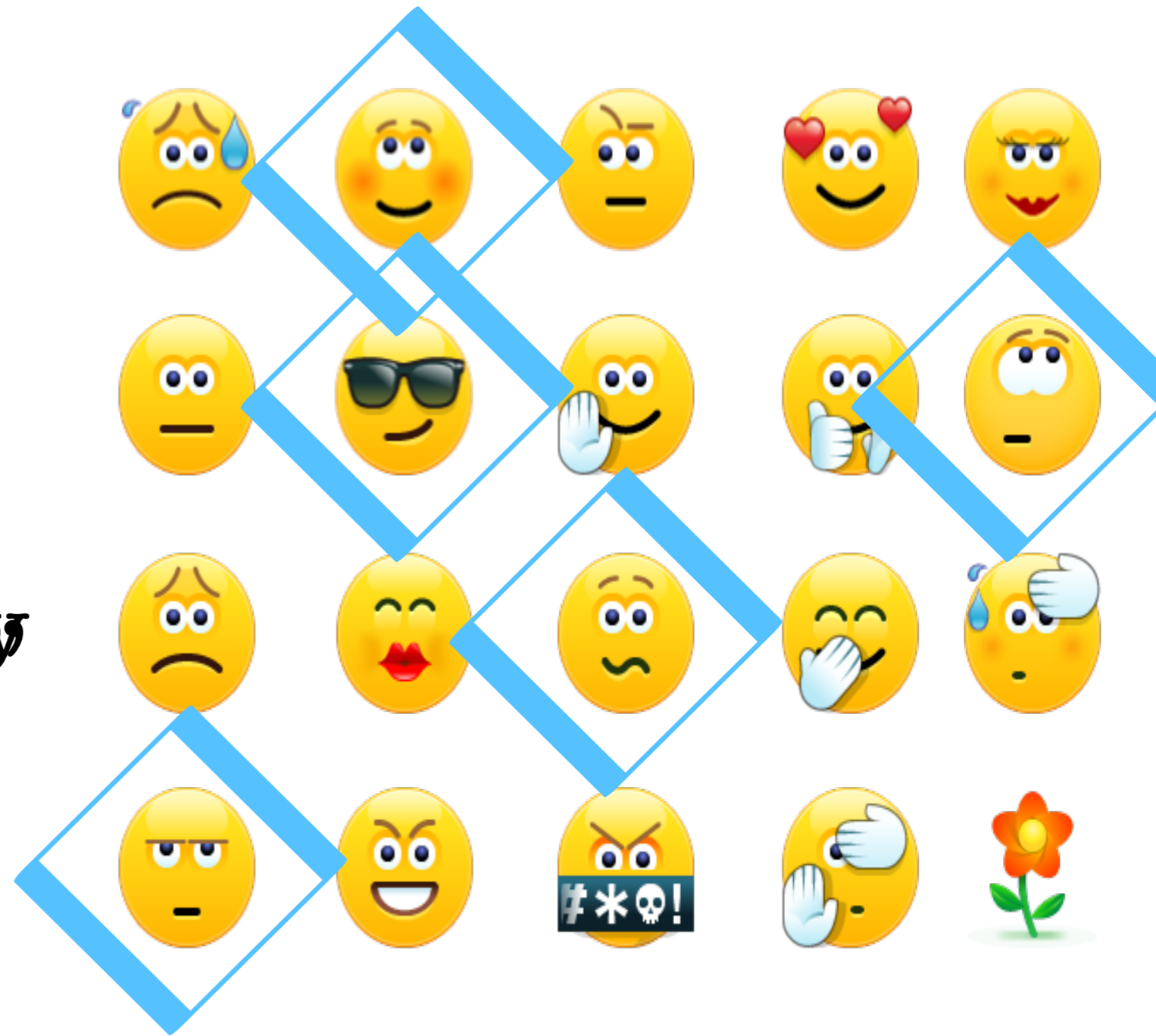


HELP ME

What's my polarity?

“I love my mobile
I'd really, truly
but I dislike
love going out in
recommend it to any
of my colleagues.”

**examples from Wikipedia*

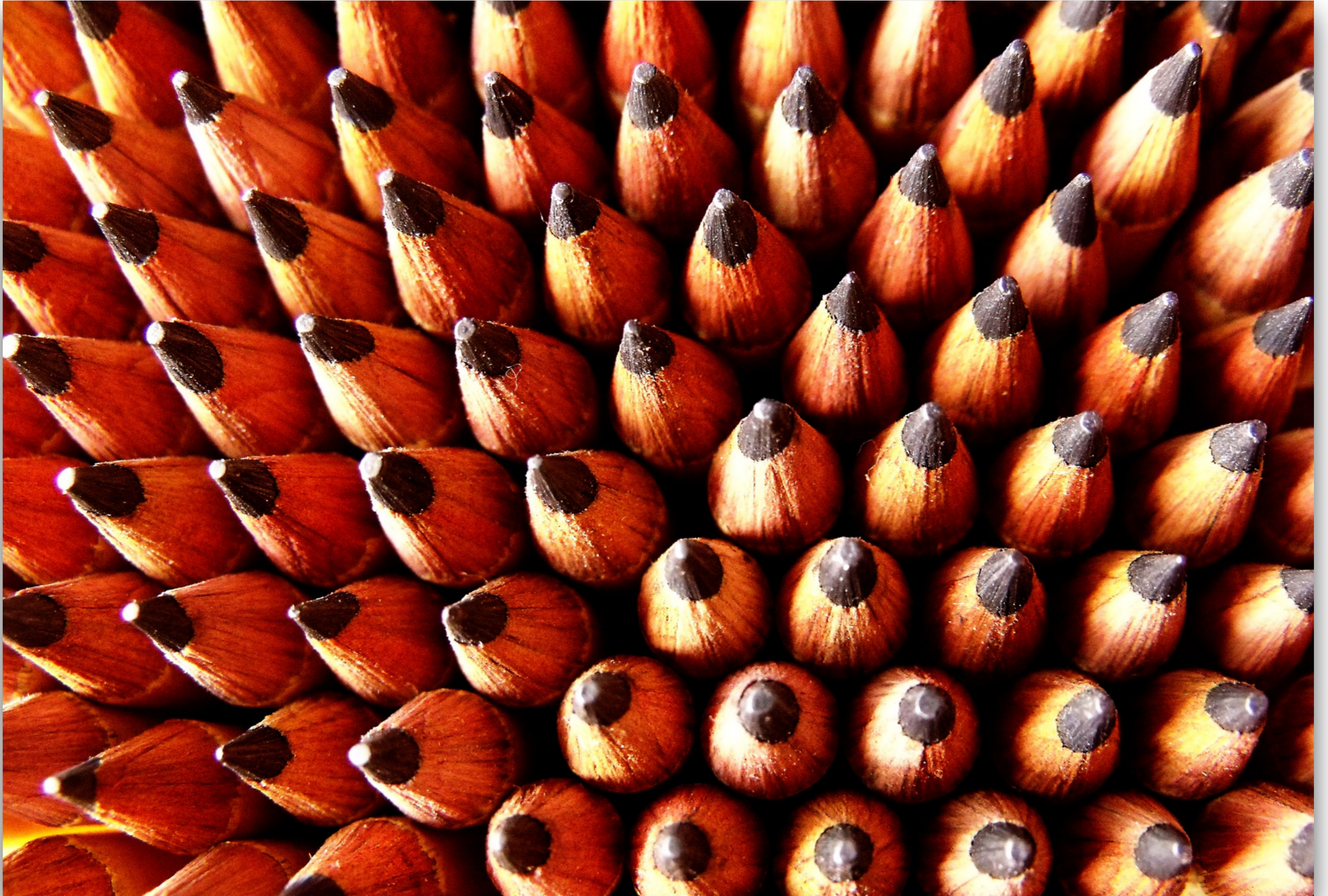


**What can I
make with it?**



@rabbigreenberg

**Sometimes you just
need to get to the...**



@rabbigreenberg

...and just tell me

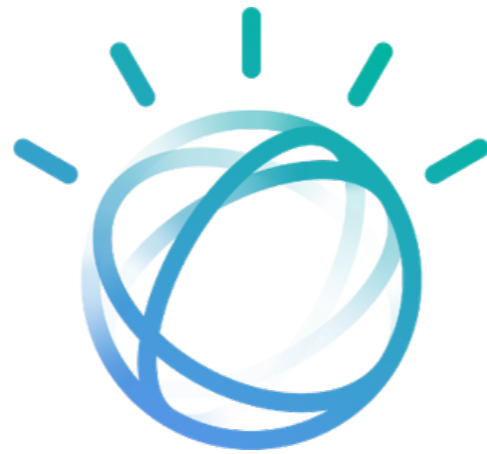
**What's the
mood of the news?**

WHAT ARE WE BUILDING

(1)



(2)



(3)



Let's build it!

@rabbigreenberg



What APIs will will be using?

NEWS API

"Search worldwide news with code"

- Generous free access tier
- Ruby SDK
- Documentation

www.newsapi.org

IBM WATSON

Natural Language Analyzer API

- Generous free access tier
- Ruby SDK
- Documentation

www.ibm.com/watson/developer

NEXMO

Messages API

- Generous free access tier
- Ruby SDK
- Documentation

developer.nexmo.com

Gemfile

```
Code File Edit Selection View Go Debug Terminal Window Help
Gemfile — mood-news
Gemfile
5
6 # Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
7 gem 'rails', '~> 5.2.3'
8 # Use postgresql as the database for Active Record
9 gem 'pg', '>= 0.18', '< 2.0'
10 # Use Puma as the app server
11 gem 'puma', '~> 3.11'
12 # Use SCSS for stylesheets
13 gem 'sass-rails', '~> 5.0'
14 # Use Uglifier as compressor for JavaScript assets
15 gem 'uglifier', '>= 1.3.0'
16 # See https://github.com/rails/execjs#readme for more supported runtimes
17 # gem 'mini_racer', platforms: :ruby
18
19 gem
20 abc gem
21
22 # Use CoffeeScript for .coffee assets and views
23 gem 'coffee-rails', '~> 4.2'
24 # Turbolinks makes navigating your web application faster. Read more: https://github.com/turbolinks
25 gem 'turbolinks', '~> 5'
26 # Build JSON APIs with ease. Read more: https://github.com/rails/jbuilder
27 gem 'jbuilder', '~> 2.5'
28 # Use Redis adapter to run Action Cable in production
```

@rabbigreenberg

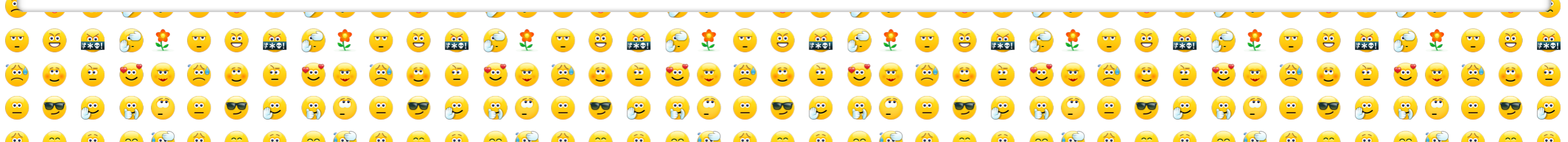
Routes



```
Code File Edit Selection View Go Debug Terminal Window Help routes.rb — mood-news  
Gemfile routes.rb nexmo.rb news_controller.r  
config > routes.rb  
1 Rails.application.routes.draw do  
2   # For details on the DSL available within this file, see http://  
3  
4 end  
5
```

@rabbigreenberg

master* 0 0 Ln 3, Col 1 Spaces: 2 UTF-8 LF Ruby



Environment Variables

☰ .env.sample

```
1 news_api_key=  
2 watson_api_key=  
3 watson_api_url=  
4 NEXMO_API_KEY=  
5 NEXMO_API_SECRET=  
6 NEXMO_WHATSAPP_NUMBER=  
7 NEXMO_APPLICATION_ID=  
8 |
```

Controller



```
Code File Edit Selection View Go Debug Terminal Window Help
news_controller.rb — mood-news
Gemfile routes.rb nexmo.rb news_controller.rb
app > controllers > news_controller.rb > ...
1 require 'news-api'
2 require 'ibm_watson/natural_language_understanding_v1'
3 require 'json'
4
5
6
7
8
9
10
11
12
13
14
15
```

Defining Methods

```
require 'news-api'
require 'ibm_watson/natural_language_understanding_v1'
require 'json'

class NewsController < ApplicationController
  include ActionView::Helpers::NumberHelper

  private

  def generate_jwt_token
    class = [
      application_id: ENV['NEWS_API_KEY']
    ]
    private_key = File.read('private_key')
    token = News::JWT.generate(token_key, pr
    token

  def get_news_headlines(topic)
    news = News.new(ENV['NEWS_API_KEY'])
    headlines = news.get_overview(topic)
    headline_string = ''
    headlines.each do |headline|
      headline_string << "#headline.title\n"
    end
    headline_string

  def analyze_headlines(topic, recipient)
    natural_language_understanding = IBM
    natural_language_understanding.url = '
    response = natural_language_understand
    text = get_news_headlines(topic)
    features = {
      "sentiment" => {},
      "keywords" => {},
      "categories" => {explanations: true
      "concepts" => {},
      "relations" => {},
      "semantic_roles" => {},
      "entities" => { "document" => true
    }
    result
    puts response

  # send whatsapp message
  send_whatsapp_msg(response, topic, rec

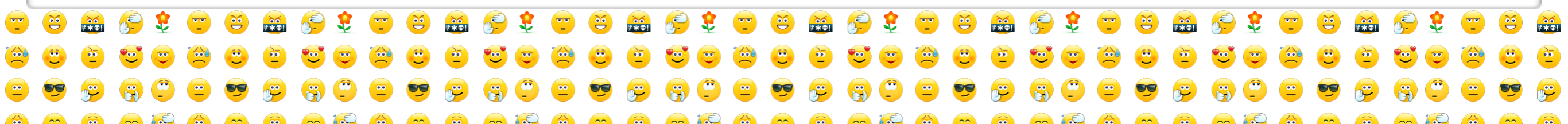
  def send_whatsapp_message(topic, recip

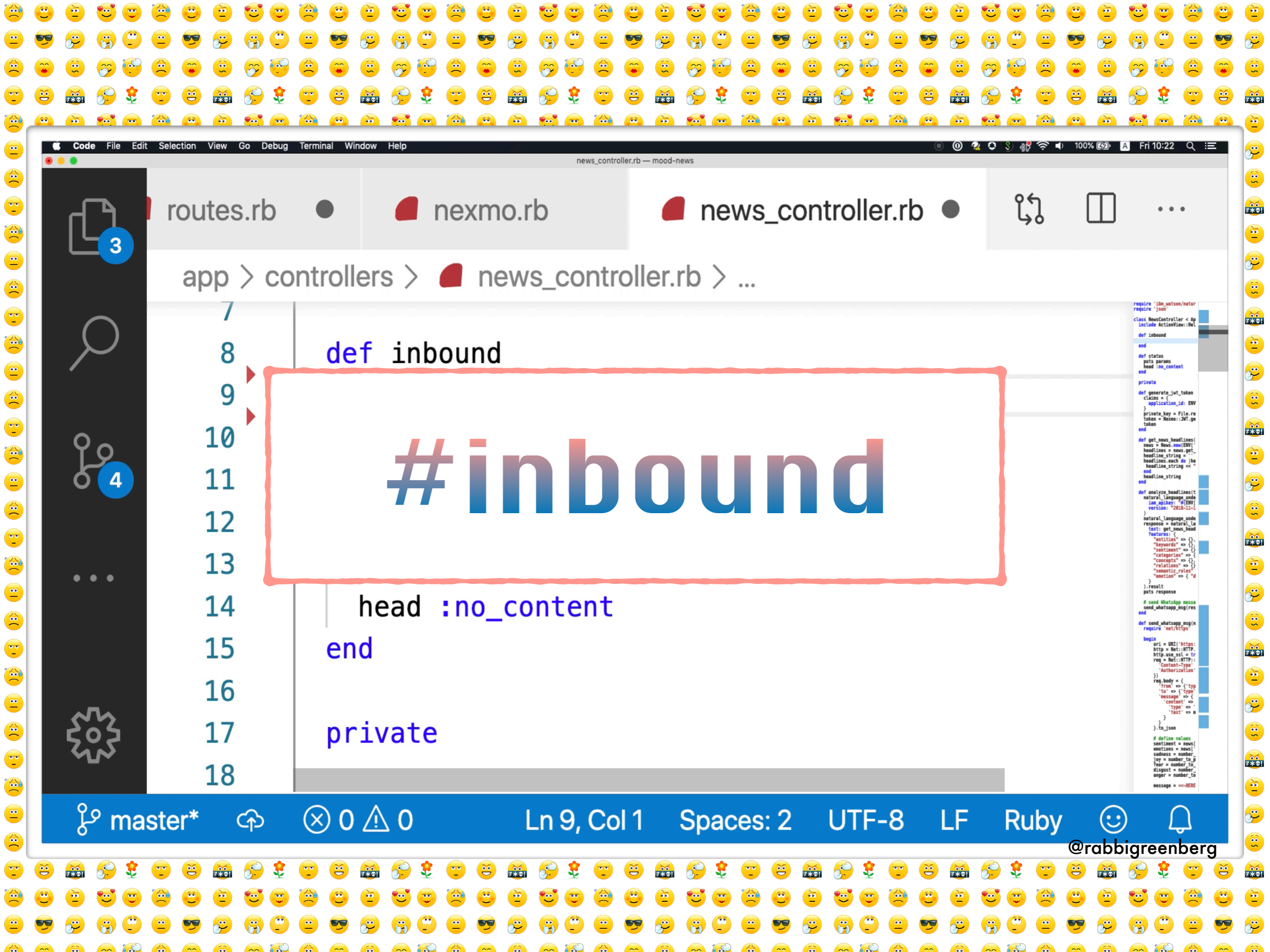
  begin
    url = URI('https://api.nexmo.com/v1
    http = Net::HTTP.new(url.host, url.p
    req = Net::HTTP::Post.new(url.path
      'Content-Type' => 'application/j
      'Authorization' => 'Bearer #tgm

    req.body = {
      'from' => { 'type' => 'whatsapp',
      'to' => { 'type' => 'whatsapp',
      'message' => {
        'content' => {
          'type' => 'text',
          'text' => message
        }
      }
    }.to_json

    # define values
    sentiment = news['sentiment'] || {}
    entities = news['entities'] || {}
    sadness = number_to_percentage(news
    joy = number_to_percentage(news[
    disgust = number_to_percentage(news
    anger = number_to_percentage(news[
    message = <<HEREDOC
    Hello there! @
    You asked for the mood of the news
    here you go!
    The overall news sentiment on #tgm
    with #sadness of sadness, #joy of
    #disgust of disgust and #anger)
    Thank you for using Pool of the M
    and the News API!
    HEREDOC
    res = http.request(req)
    puts "response: #{res.body}"
    puts message
    puts JSON.parse(res.body)
    rescue => e
    puts "failed: #{e}"
  end
end
```

@rabbigreenberg





```
Code File Edit Selection View Go Debug Terminal Window Help
news_controller.rb — mood-news
routes.rb nexmo.rb news_controller.rb
app > controllers > news_controller.rb > ...
7
8 def inbound
9
10
11 # inbound
12
13
14 head :no_content
15 end
16
17 private
18
```

Ln 9, Col 1 Spaces: 2 UTF-8 LF Ruby

@rabbigreenberg



```
Code File Edit Selection View Go Debug Terminal Window Help
news_controller.rb — mood-news

routes.rb ● nexmo.rb news_controller.rb ●

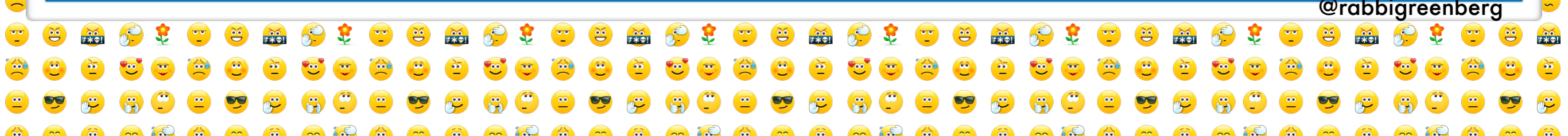
app > controllers > news_controller.rb > ...

44
45
46
47
48
49
50
51
52
53
54
55
```

```
recipient_number = pa
# get_headlines and p
analyze_headlines(top
head :no_content
end
def status
puts params
head :no_content
end
private
def generate_jwt_token
claims = {
application_id: ENV
}
private_key = File.re
token = Nexmo::JWT.ge
token
end
def get_news_headlines(
news = News.new(ENV[
headlines = news.get_
headline_string = {
headlines.each do |h
headline_string <<
end
headline_string
end
end

def send_whatsapp_msg(n
require 'net/http'
begin
uri = URI('https:
http = Net::HTTP
http.use_ssl = tr
req = Net::HTTP::
'Content-Type'
'Authorization'
})
req.body = {
from => {'typ
'to' => {'type
'message' => {
'content' => {
'type' => '
'text' => m
}
}.to_json
# define values
sentiment = news[
emotions = news[
sadness = number_
joy = number_to_p
fear = number_to_
disgust = number_
anger = number_to
message = <<HERE
Hello there! W
You asked for the
Here you go!
The overall news
with #{sadness} a
#{disgust} of dis
Thank you for usi
and the News API!
HEREDOC
res = http.request
puts "response #{
```

#analyze_headlines



#get_news_headlines

34
35

41
42
43
44
45

```
)  
natural_language_understanding.url = 'https://ga  
response = natural_language_understanding.analyze  
text: get_news_headlines(topic),  
features: {
```

```
topic = params[:messa  
recipient_number = pa  
# get headlines and p  
analyze_headlines(top  
head :no_content  
end  
def status  
puts params  
head :no_content  
end  
private  
def generate_jwt_token  
claims = {  
  application_id: ENV  
}  
private_key = File.re  
token = Nexmo::JWT.ge  
token  
end  
def analyze_headlines(t  
natural_language_unde  
iam_apikey: #ENV[  
version: "2018-11-1  
}  
natural_language_unde  
response = natural_la  
text: get_news_head  
features: {  
  "entities" => {},  
  "keywords" => {},  
  "sentiment" => {},  
  "categories" => {},  
  "concepts" => {},  
  "relations" => {},  
  "semantic_roles"  
  "emotion" => {}  
}  
}.result  
puts response  
# send WhatsApp messa  
send_whatsapp_msg(res  
end  
def send_whatsapp_msg(n  
require 'net/http'  
begin  
  uri = URI('https:  
  http = Net::HTTP,  
  http_use_ssl = tr  
  req = Net::HTTP:::  
  'Content-Type'  
  'Authorization'  
})  
  req.body = {  
    'from' => { 'typ  
    'to' => { 'type'  
    'message' => {  
      'content' => {  
        'type' => :  
        'text' => m  
      }  
    }, :to_json  
  }  
  # define values  
  sentiment = news!  
  emotions = news!  
  sadness = number,  
  joy = number, to_p  
  fear = number, to  
  disgust = number,  
  anger = number, to  
  message = <<HERE  
  Hello there! W  
  You asked for the  
  Here you go!  
  The overall news  
  with #(sadness) o  
  #(disgust) of dis
```

```
require 'net/https'
```

```
begin
```

```
  uri = URI('https://api.nexmo.com/v0.1/messages')
```

```
  http = Net::HTTP.new(uri.host, uri.port)
```

```
  http.use_ssl = true
```

```
  req = Net::HTTP::Post.new(uri.path, {
```

```
  })
```

```
  re
```

#send_whatstapp_msg

```
  'message' => {
```

```
    'content' => {
```

```
      'type' => 'text',
```

```
      'text' => message
```

```
    }
```

```
  }
```

```
}
```

```
}.to_json
```

@rabbigreenberg


```
# define values
```

```
sentiment = news['sentiment']['document']['label']
```

```
emotions = news['emotion']['document']['emotion']
```

```
sadnes
```

```
joy =
```

```
fear =
```

```
disgus
```

```
anger = number_to_percentage(emotions['anger'] * 100)
```

```
message = <<~HEREDOC
```

@rabbigreenberg

#send_whatstapp_msg

```
message = <<~HEREDOC
```

```
Hello there! 🙌
```

```
You asked for the mood of the news on the following topic: #{topic}.  
Here you go!
```

```
The over  
with #  
#{disgu
```

#send_whatapp_msg

```
Thank you for using mood of the news powered by NEXMO, IBM watson  
and the News API!  
HEREDOC
```

```
res = http.request(req)
```

```
puts "response #{res.body}"
```

@rabbigreenberg

app > controllers > news_controller.rb > ...

25

26

```
def generate_jwt_token
```

27

```
  claims = {
```

28

29

```
#generate_jwt_token
```

30

31

```
  token = JWT.generate(claims, private_key)
```

32

```
  token
```

33

```
end
```

34

@rabbigreenberg

Giving It A Go!



Nexmo WhatsApp



8/1/2019

Messages to this chat and calls are secured with end-to-end encryption. Nexmo WhatsApp may use another company to store, read and respond to your messages and calls. Click for more info.



Type a message



Thank you!

 @rabbigreenberg



<https://github.com/Nexmo/mood-of-the-news>