

VOXP

GENERATIVE GRAMMARS IN ELIXIR

**LET'S TALK
ABOUT TEXT.**



Appropriate Tributes

@godtributes



Following

@hello_skyler VERBS FOR THE VERB GOD!
TRANSITIVES FOR THE TRANSITIVE THRONE

6:53 AM - 4 May 2016





BotPointer

@BotPointer

It feels good to be good at all.

LIKE

1



4:05 PM - 27 Apr 2016





BotPointer

@BotPointer

ProTip: Never go back to full hipster.

RETWEETS

3

LIKES

3



3:10 PM - 30 Mar 2016





BotPointer

@BotPointer

 **Follow**

Time to go and withdraw cash from the bank.

5:30 AM - 30 Apr 2016





BotPointer

@BotPointer

Rubyists, how does it feel to be that guy who starts everyone clapping between movements at the symphony?

RETWEET

1

LIKES

4



1:27 AM - 11 Mar 2016





BotPointer

@BotPointer

That exciting moment when you have it in your hand, but every piece is hard fought.

5:15 AM - 26 Mar 2016





BotPointer

@BotPointer

If anyone is interested, I'm already starting to feel negative about your product.

RETWEET

1

LIKE

1



9:59 PM - 7 Mar 2016





piecritic_ebooks

@piecritic_ebook



Follow

I am required by law to tell you however that I am not looking forward to how much of my tax has been lit on fire by the Liberals

RETWEETS

4

LIKES

2



9:11 AM - 27 Apr 2016



**GARBAGE IN,
GARBAGE OUT.**



**OKAY. IF NOT
TWITTER OR CHAT
THEN WHERE?**



UNGUIDED TEXT GENERATION == BAD.

TEMPLATES

SMALL TALK, NOT SMALLTALK™

{greeting} {observation} {opinion}

"Hey there! SpaceX finally landed the Falcon 9 first stage on the floating barge. Science is incredible."

GREETINGS

```
rule :greeting, [  
    "Did you hear?",  
    "Good news everybody!",  
    "O. M. G.",  
    "Have you heard?",  
    "Fresh from the newswire:"  
]
```

OBSERVATIONS

```
rule, :observation, [  
  "SpaceX finally landed the Falcon 9 first stage on the floating barge.",  
  "Ecto 2 looks like it will bring a lot of improvements.",  
  "The Giro D'Italia starts on Friday.",  
  "A sportsball team had an unlikely victory this week.",  
  "The hex website has had a nice update.",  
  "It turns out the internet will ruin everything.",  
  "Donald Trump has a non-zero chance of becoming president",  
]
```

OPINIONS

```
rule :opinion, [  
    "Science is wild.",  
    "This is the final portent of the end times.",  
    "There's never been a more exciting time to be a chatbot.",  
    "I don't know what to do with myself.",  
    "I'm as surprised as you are."  
]
```

SURPRISE

DSLICIOUS

```
defmodule Smalltalk.Starter do
  use Voxpop.Grammar

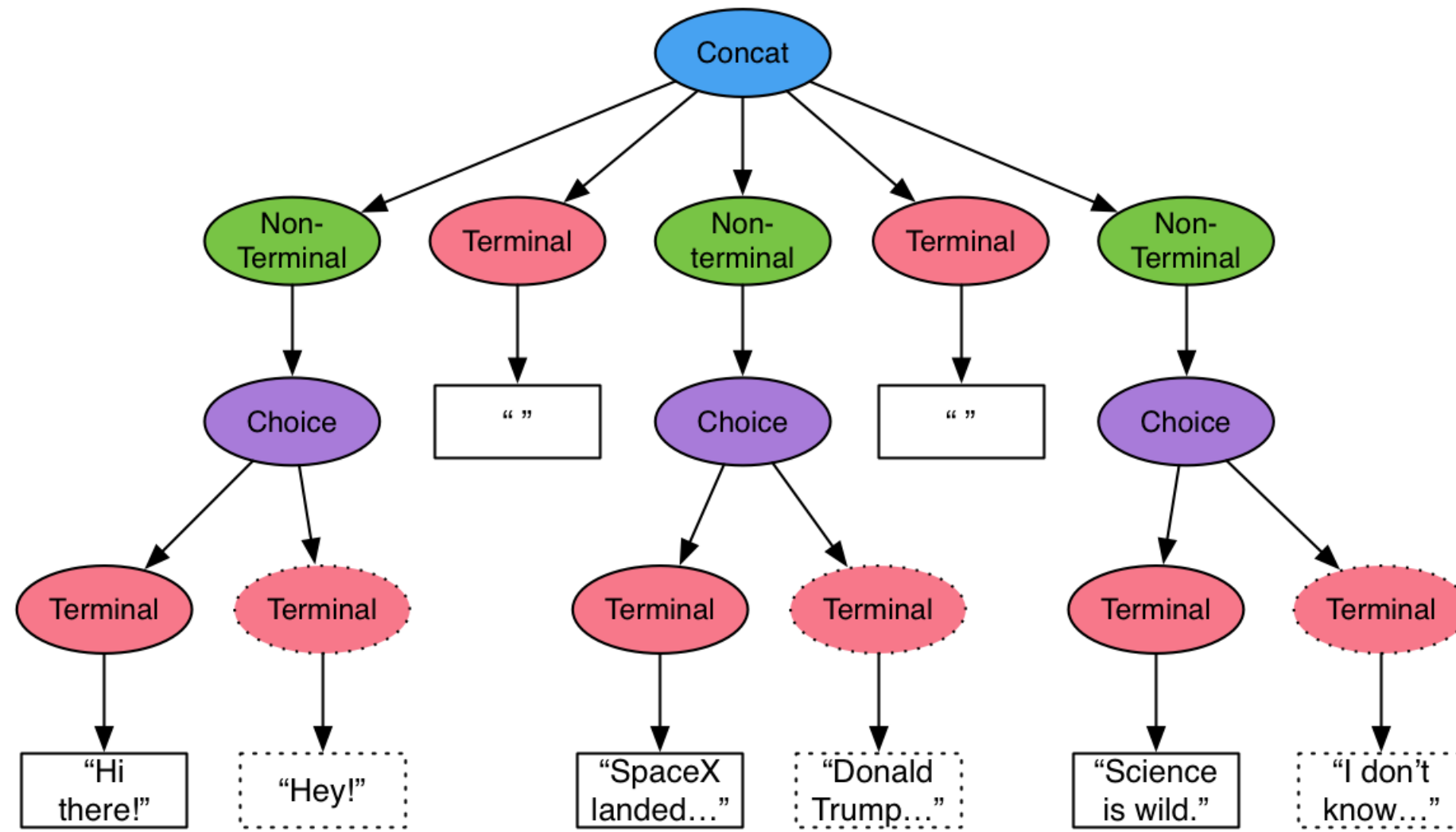
  start "{greeting} {observation} {opinion}"

  rule :greeting, [
    "Did you hear?",
    "Good news everybody!",
    "O. M. G.",
    "Have you heard?",
    "Fresh from the newswire:"
  ]

  rule :observation, [
    "SpaceX finally landed the Falcon 9 first stage on the floating barge.",
    "Ecto 2 looks like it will bring a lot of improvements.",
    "The Giro D'Italia starts on Friday.",
    "A sportsball team had an unlikely victory this week.",
    "The hex website has had a nice update.",
    "It turns out the internet will ruin everything.",
    "Donald Trump has a non-zero chance of becoming president.",
  ]

  rule :opinion, [
    "Science is wild.",
    "This is the final portent of the end times.",
    "There's never been a more exciting time to be a chatbot.",
    "I don't know what to do with myself.",
    "I'm as surprised as you are.",
    "I don't want to live anymore."
  ]
end
```

HOLY AST BATMAN



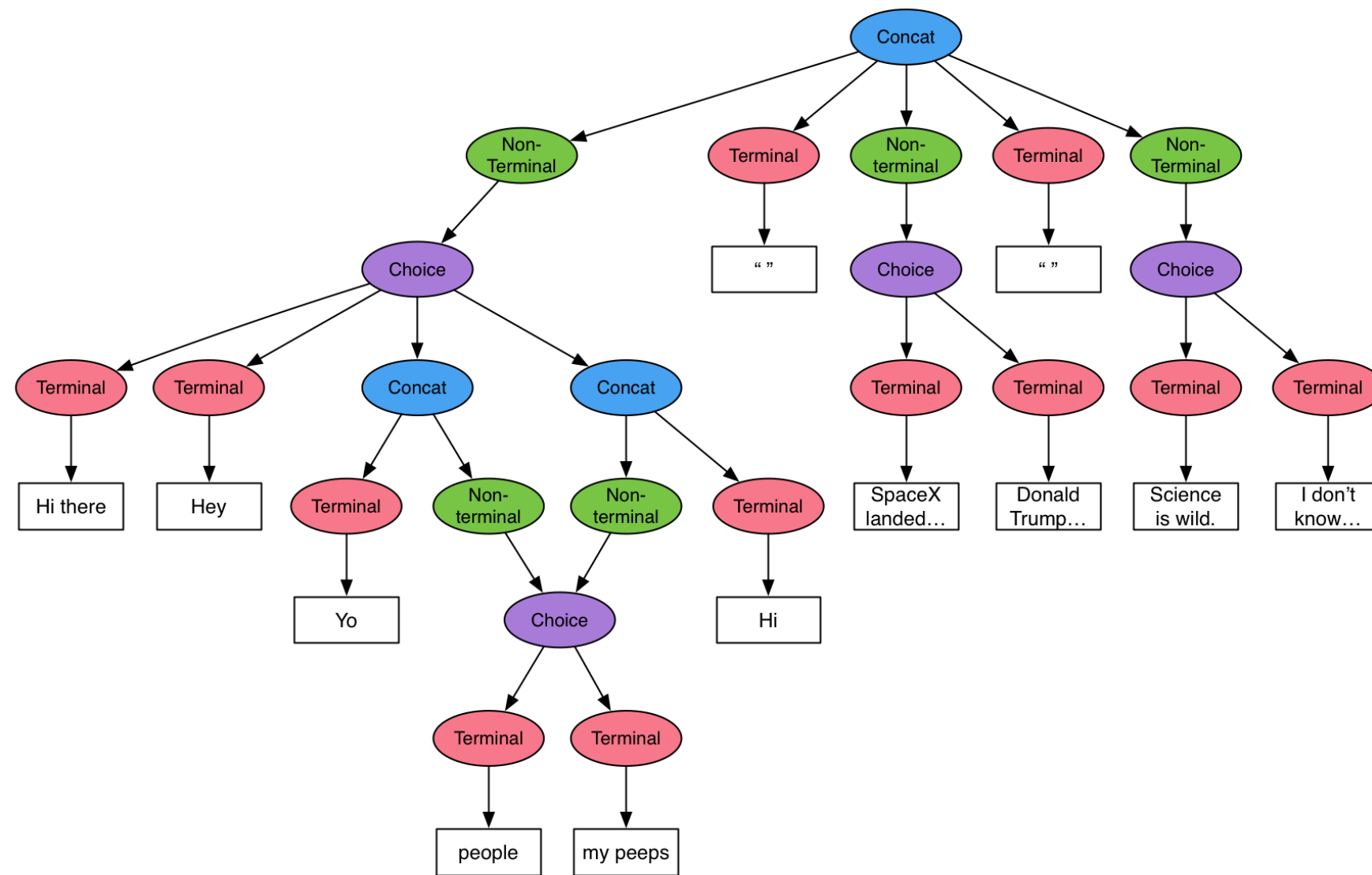
"Fresh from the newswire: SpaceX finally landed the Falcon 9 first stage on the floating barge. I'm as surprised as you are."

"Did you hear? The hex website has had a nice update. There's never been a more exciting time to be a chatbot."

"Did you hear? Ecto 2 looks like it will bring a lot of improvements. This is the final portent of the end times."

"O. M. G. The hex website has had a nice update. I'm as surprised as you are."

MOAR OPTIONS!



MOAR OPTIONS!

```
rule :greeting, [  
    "Did you hear?",  
    "Good news {people}!",  
    "Yo {people}!",  
    "Wake up {people}!",  
    "Have you heard?",  
    "Fresh from the newswire:"  
]
```

MOAR OPTIONS!

```
rule :people, [  
    "comrades",  
    "friends",  
    "team",  
    "humans",  
    "my peeps",  
    "good people"  
]
```

"Fresh from the newswire: The hex website has had a nice update. This is unprecedented."

"Wake up comrades! The hex website has had a nice update. Wowzers."

"Wake up friends! The Giro D'Italia starts on Friday. There's never been a more exciting time to be a chatbot."

PORTED FROM RUBY
GITHUB.COM/MAETL/CALYX

NOW FOR SOME ELIXIR
GITHUB.COM/ZOVA/VOXPOP

PRODUCTION

```
defprotocol Voxpop.Production do
  def evaluate(production, registry)
end
```

TERMINAL

```
defmodule Voxpop.Terminal do
  defstruct atom: ""

  defimpl Voxpop.Production do
    def evaluate(terminal, _registry) do
      terminal.atom
    end
  end
end
```

NON-TERMINAL

```
defmodule Voxpop.NonTerminal do
  defstruct :rule
end

defimpl Voxpop.Production, for: Voxpop.NonTerminal do
  def evaluate(non_terminal, registry) do
    registry.rules
    |> Map.get(non_terminal.rule)
    |> Voxpop.Production.evaluate(registry)
  end
end
```

CONCAT

```
defmodule Voxpop.Concat do
  defstruct parts: []

  def parse(rule, matches) do
    %Voxpop.Concat{parts: do_parse(rule, List.flatten(matches), 0)}
  end

  defp do_parse(rule, [], offset) do
    [wrap_text(binary_part(rule, offset, byte_size(rule) - offset))]
  end

  defp do_parse(rule, [{pos, length} | matches], offset) do
    keep = pos - offset
    new_offset = pos + length

    << _ :: binary-size(offset), part::binary-size(keep), token::binary-size(length), _::binary>> = rule
    [wrap_text(part) | [extract_key(token) | do_parse(rule, matches, new_offset)]]
  end
end
```

(contd.)

CONCAT

```
defp extract_key(token) do
  key = token |> String.slice(1..-2) |> String.to_atom
  %Voxpop.NonTerminal{rule: key}
end

defp wrap_text(part) do
  %Voxpop.Terminal{atom: part}
end

defimpl Voxpop.Production, for: Voxpop.Concat do
  def evaluate(concat, registry) do
    concat.parts |> Enum.map(&Voxpop.Production.evaluate(&1, registry)) |> Enum.join
  end
end
```

WHAT NOW?

WEIGHTED CHOICE

A CHOICE WHICH IS SKEWED TOWARDS PARTICULAR OPTIONS

```
rule :feeling,[  
    awful: 0.15  
    bad: 0.2  
    okay: 0.4  
    good: 0.2  
    great: 0.15  
]
```

CONTEXT

INJECTING RUNTIME DATA INTO A PRE-GENERATED GRAMMAR DEFINITION

```
MyGrammar.generate(%{  
  name: "Andrew",  
  preferred_greeting: "Hello {name}"  
})
```

TRANSFORMS

RUN FUNCTIONS ON EVALUATED STRINGS

```
start "{greeting|capitalize}, {noun|pluralize}"
```

PER GENERATION REPETITION AVOIDANCE

**IF A CHOICE RULE IS USED TWICE IN A GENERATE, AVOID
REPEATING THE SAME CHOICE**

"What a {positive_word} day today! Aren't you feeling {positive_word}?"

INTER GENERATION REPETITION AVOIDANCE

**THE SAME AS ABOVE, BUT AVOID REPEATING YOURSELF
ACROSS GENERATIONS**

- _ (ツ) _ / -

**API DESIGN BY
SHOUTING!**

QUESTIONS?

I HAVE BEEN @MOOTPOINTER