

# Data Science Salon Miami November 2018

## Workshop Presentation Notes

Anton Antonov, Accendo Data LLC

November 6, 2018

# Outline

# Opening

- ▶ Mission statement.
  - ▶ **Primary:**
    - ▶ To introduce, teach, and illustrate the making of goal oriented conversational agents using context free grammars and finite state machines .
  - ▶ **Secondary:**
    - ▶ To speed up the coming of the next AI winter .
- ▶ The GitHub repository.

# What we are going to do?

- ▶ What kind of conversational agents we consider here?
  - ▶ For machine learning workflows.
  - ▶ For super-user or end-user interfacing of software components.
  - ▶ Hence (most, but not all) of the conversational agents that are shallow or not conversational much.
    - ▶ Shallow: one or two states at most.
  - ▶ But yes, this can be used for "deeper meaning" conversational agents.
- ▶ Other uses:
  - ▶ for brainstorming;
  - ▶ for project management;
  - ▶ for Domain Specific Languages (DSL's).

# Who am I?

## Education

- ▶ MSc Computer Science (Data Bases)
- ▶ MSc Mathematics (Abstract Algebra)
- ▶ PhD Applied Mathematics (Large Scale Air-Pollution Simulations)

## General point of view on AI

- ▶ Siding with the Weak AI.
- ▶ See the related blog post and panel video from the Data Science Salon Miami February 2018.
  - ▶ [TowardData Science post link](#)

# The companies I have made conversational agents for

- ▶ Panasonic Automotive Systems America
  - ▶ *butler in the car*
- ▶ Entefy
  - ▶ *universal communicator*
- ▶ Clearsense LLC
  - ▶ *diabetics management, patient critical conditions predictions*
- ▶ Christy Health Inc
  - ▶ *patient critical conditions prediction, data transformations*
- ▶ AITO Consulting LLC
  - ▶ *accounting data handling and cash flow prediction*
- ▶ QiO Technologies
  - ▶ *predictions and recommendations in Industry 4.0*

# In case you hear things you not like

- ▶ I guarantee you many of the employers I engage(d) with would not like everything I plan to say.
- ▶ It is like eating vegetables:
  - ▶ even if you do not like them they are good for you.
- ▶ If you do not like what you hear please write me an (extensive) email about how wrong I am:
  - ▶ [antononcube@gmail.com](mailto:antononcube@gmail.com)

## Managing expectations

- ▶ You are not going to learn how to make a complete, say, Alexa skill in 2 hours.
  - ▶ Or 20 if this is the first time you see this kind of expositions.
- ▶ In this workshop we only sketch making of complete Alexa skills or Google Home apps/skills.
  - ▶ Note that there are a lot of other conversational platforms.
- ▶ What are you going to see is a particular type of conversational agent making based on grammars and finite state machines.
- ▶ Because of what Amazon, Apple, Google, Nuance, and others developments of speech-to-text recognition frameworks we are taking for granted to be easily able to hook-up with a certain speech-to-text module (or two, or five.)
  - ▶ Same for text-to-speech.
- ▶ Note that there are a plenty of dialog system design paradigms / approaches.
- ▶ I have a script for only 40 min, the rest of the workshop is unscripted.
  - ▶ What we are going to do would depend on:
    - ▶ what kind of background the audience has, and
    - ▶ what kind of agents the audience wants to design.



# GitHub project for the workshop and installations required

- ▶ The GitHub project:
  - ▶ `https://github.com/antononcube/ConversationalAgents/tree/master/Projects/DataScienceSalon-Miami-Nov-2018-Workshop`
  - ▶ Or just find it in:
    - ▶ `https://github.com/antononcube/ConversationalAgents/`
- ▶ While I am talking you can (try to) install the software tools listed in that repository.
  - ▶ Rakudo (for Perl6.)
    - ▶ Up to you; better use the online tool:  
`https://glot.io/new/perl6`
  - ▶ Atom editor (for Perl6 and Python.)
    - ▶ Install the corresponding packages too.
  - ▶ R and RStudio (for R.)
  - ▶ Mathematica (used, but not needed.)
  - ▶ ANTLR (listed, but not needed.)
    - ▶ *I would rather use Perl6 at this point.*

# The screen set-up

- ▶ Four screens:
  - ▶ Mind-map,
  - ▶ The main presentation,
  - ▶ The Atom editor/IDE,
  - ▶ RStudio.

# Opening examples

## Didactic

- ▶ ☒ Phone Dialogs Conversational Agent.
- ▶ ☐ Love food
  - ▶ Simple we are going to download it, use it, and extend it 5-10 min from now.

## Eliza run example

# Shock and awe examples (I hope...)

- ▶ ☒ Regression workflows
  - ▶ QRMon main workflow example.
  - ▶ Note the three-four regression methods presented:
    - ▶ Quantile Regression;
    - ▶ Linear Regression;
    - ▶ Neural Networks regression.
- ▶ ☐ Classification workflows.
  - ▶ CICon main workflow.
  - ▶ Rapid creation of classification workflows.

# Complex Conversational Agent example

- ▶ Diabetes management: Glukoza.
- ▶ Note the complex nature of the design.
- ▶ Two perspectives are accommodated:
  - ▶ "simple" end user, and
  - ▶ physician / researcher.

# The big picture

- ▶ In this workshop we concentrate on first ~~four~~ six steps in the following workflow.
- ▶ The main workflow simplified:
  1. Get and brainstorm on an automation idea.
  2. Gather or come-up with dialogs.
  3. Make suitable grammars / DSL's.
  4. Come up with finite states and transition between them.
  5. Program parser(s).
  6. Program interpreter(s).
  7. Refine with initial feedback.
  8. Decide when to stop.
- ▶ The use of monadic DSL's big picture.

## "Why not?" discussion points

- ▶ It is very important to know the limitations of your tools.
- ▶ *In case the audience is interested to hear this and we have time.*

Why not Java?

Why not JavaScript?

Why not Mathematica?

Why not Perl6?

Why not Python?

Why not R?

# Introduction to EBNF

- ▶ Context free grammars.
  - ▶ A set of production rules.
  - ▶ You will know it when you see it.
  - ▶ Comes from Noam Chomsky's formal grammars hierarchy.
- ▶ Extended Backus-Naur Form.



# Introduction to parser programming

- ▶ This is not that important to follow.
- ▶ Actual parser programming in R.
- ▶ Should we do it *also* in Perl6? (coming up next. . . )
  - ▶ I think yes.
- ▶ Contrasting the two approaches.
  - ▶ Functional parsers vs
  - ▶ Declarative rules.

# Introduction to parser generation

- ▶ Here we use the declarative rules.
- ▶ Languages
  - ▶ Perl6
  - ▶ Python
  - ▶ Mathematica
  - ▶ sorry, no R.
- ▶ Grammar inclusions and reuse.

# Interpretation

- ▶ What if you parsing tree is also code?
  - ▶ Lisp, Mathematica
- ▶ What if parsing tree traversal is baked-in into the language?
  - ▶ Perl6, Scala
- ▶ Alternatively, you can get code that traverse the tree.
  - ▶ ANTLR

# Grammar making exercises

## Love food grammar

- ▶ Add more food items and check can you parse sentences with them.
- ▶ Add new verbs.
- ▶ Add new commands. E.g.
  - ▶ Where to find the best ...?
- ▶ What other actions to hook-up?
  - ▶ (Instead of just gain calories.)

## dplyr natural language command

- ▶ What other commands to add?
- ▶ What other functionalities to program for the existing commands?

# Break

# Deciding what conversational agent to design

- ▶ Natural language commands for dplyr.
- ▶ Will they kill me?
  - ▶ I have a half-baked interactive demo dashboard.
- ▶ Regression workflows.
  - ▶ Fully developed.
- ▶ Job search.
- ▶ Movie search and recommendations.
- ▶ Construction and training of neural networks.

# Gather dialogs

- ▶ How are we going to gather the dialogs?
  - ▶ By typing in?
  - ▶ By a public Slack channel?
    - ▶ [#datasciencesalon.slack.com](https://datasciencesalon.slack.com) #conversational-agents
  - ▶ By email: [antononcube@gmail.com](mailto:antononcube@gmail.com)

# Making Morphological Analysis tables

- ▶ Morphological Analysis is used for problem solving.
- ▶ Consider:
  - ▶ multi-dimensional, non-quantified complex problems.
  - ▶ open-ended problems,
  - ▶ wicked problems.
- ▶ Dealing with seemingly non-reducible complexity.
- ▶ Made by Fritz Zwicky for star classification, etc.



# Describe and program grammars

- ▶ ☐ Perl6
  - ▶ It is very likely I would use Perl6.
- ▶ ☐ Python
- ▶ ☐ Mathematica

# Generation of parsers

- ▶ ☐ Perl6
- ▶ ☐ Python
- ▶ ☐ Mathematica

# Conclusion

Why keep learning about this?

Where to go next?

# References

1. Anton Antonov, Creating and programming domain specific languages, (2016), [MathematicaForPrediction at WordPress](#).
2. Lars Dieckow, "Parsing with grammars shoot-out", (2017), [The Perl Conference](#).