

Chatbots and Digital Assistants

Tutorial sobre RASA Hello Project

19/09/2022

RASA Framework

www.rasa.com



- **Free Open source** framework para conversas de texto automatizadas. Permite entender mensagens, manter conversas e conectar-se a canais de mensagens e APIs (E-schooling).
- Permite construir um **sistema de diálogo orientado a tarefas**;
- Quase nenhuma habilidade de programação é necessária para trabalhar com RASA.
- Esta ferramenta atende aos requisitos do projeto Hello;

RASA Framework



Rasa

Working with many notable brands – including T-Mobile, Orange, and Adobe – Rasa offers an open-source toolkit of Conversational AI solutions. As a result, Gartner recommends that companies with first-rate software engineering and application development capabilities consider the vendor. After all, it provides the ideal platform for deep customizations. However, a highly-skilled team of developers is vital to benefit fully from the solution.

[Fonte](#)

Yet Another Markup Language (YAML)

A value:

```
value
```

A value named “foo”:

```
foo: value
```

Scalar types

```
a: 1      # integer
a: 1.234  # float
b: 'abc'  # string
b: "abc"
b: abc
c: false  # boolean type
d: 2015-04-05 # date type
```

A list:

```
- 1
- 2
- 'a string'
```

A list named “bar”:

```
bar:
  - 1
  - 2
  - 'a string'
```

Alternate list syntax (“Flow style”):

```
bar: [1, 2, 'a string']
```

- [YAML 1.2 Spec](#)
- [Online YAML Parser \(Python\)](#)

A dictionary:

```
key1: val1
key2: val2
key3: val3
```

A dictionary named “joe”:

```
joe:
  key1: val1
  key2: val2
  key3: val3
```

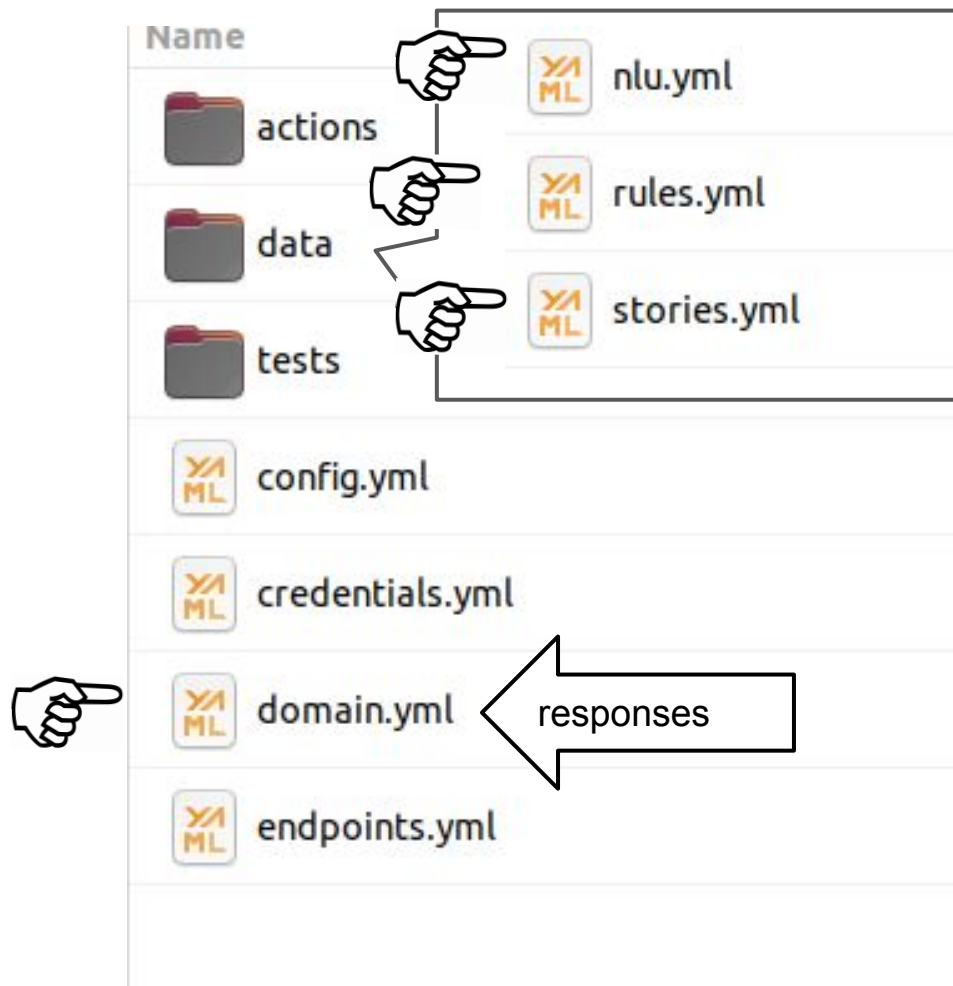
Dictionary in flow style:

```
joe: {key1: val1, key2: val2, key3: val3}
```

A list of dictionaries:

```
children:
  - name: Jimmy Smith
    age: 15
  - name: Jenny Smith
    age: 12
```

Estrutura Geral



Command	Effect
<code>rasa init</code>	Creates a new project with example training data, actions, and config files.
<code>rasa train</code>	Trains a model using your NLU data and stories, saves trained model in <code>./models</code> .
<code>rasa interactive</code>	Starts an interactive learning session to create new training data by chatting to your assistant.
<code>rasa shell</code>	Loads your trained model and lets you talk to your assistant on the command line.
<code>rasa run</code>	Starts a server with your trained model.
<code>rasa run actions</code>	Starts an action server using the Rasa SDK.
<code>rasa visualize</code>	Generates a visual representation of your stories.
<code>rasa test</code>	Tests a trained Rasa model on any files starting with <code>test_</code> .
<code>rasa data split nlu</code>	Performs a 80/20 split of your NLU training data.
<code>rasa data convert</code>	Converts training data between different formats.
<code>rasa data migrate</code>	Migrates 2.0 domain to 3.0 format.
<code>rasa data validate</code>	Checks the domain, NLU and conversation data for inconsistencies.
<code>rasa export</code>	Exports conversations from a tracker store to an event broker.
<code>rasa evaluate markers</code>	Extracts markers from an existing tracker store.
<code>rasa x</code>	Launches Rasa X in local mode.
<code>rasa -h</code>	Shows all available commands.

NLU file

NLU = Natural language Understanding

```
- intent: affirm
  examples: |
    - sim
    - s
    - claro
    - isso parece-me bem
    - correto
- intent: deny
  examples: |
    - não
    - n
    - nunca
    - Não me parece
    - Eu não gosto
    - nem pensar
    - NÃO
```

Intent:

Captura o sentido geral da frase e classificar;

Entity:

Aspetos relevantes do input do utilizador que podem ser usados posteriormente na conversa.

```
nlu:
- intent: inform
  examples: |
    - My account number is [1234567890](account_number)
```

Response:

Define as respostas do bot.

```
responses:  
  utter_greet:  
    - text: Ola! Em que posso ajudar?  
  utter_submit:  
    - text: O teu pedido foi enviado ao docente.  
  utter_slots_values:  
    - text: O professor {prof_name} vai entrar em contacto através do email {email}!
```

```
intents:  
  - greet  
  - deny  
  - request_help  
  - goodbye  
  - affirm  
  - mood_great  
  - mood_unhappy  
  - bot_challenge  
  - thanks
```

Stories and Rules Files

Story:

- Define os caminhos que a conversa pode seguir;
- Exemplo para aprendizagem;

```
- story: happy path
  steps:
  - intent: greet
  - action: utter_greet
  - intent: mood_great
  - action: utter_happy
```

Rules:

- Define partes da conversa que seguem sempre o mesmo padrão.
- Padrão que o bot deve seguir;

```
rules:
- rule: Say goodbye anytime the user says goodbye
  steps:
  - intent: goodbye
  - action: utter_goodbye
```


Slots and forms

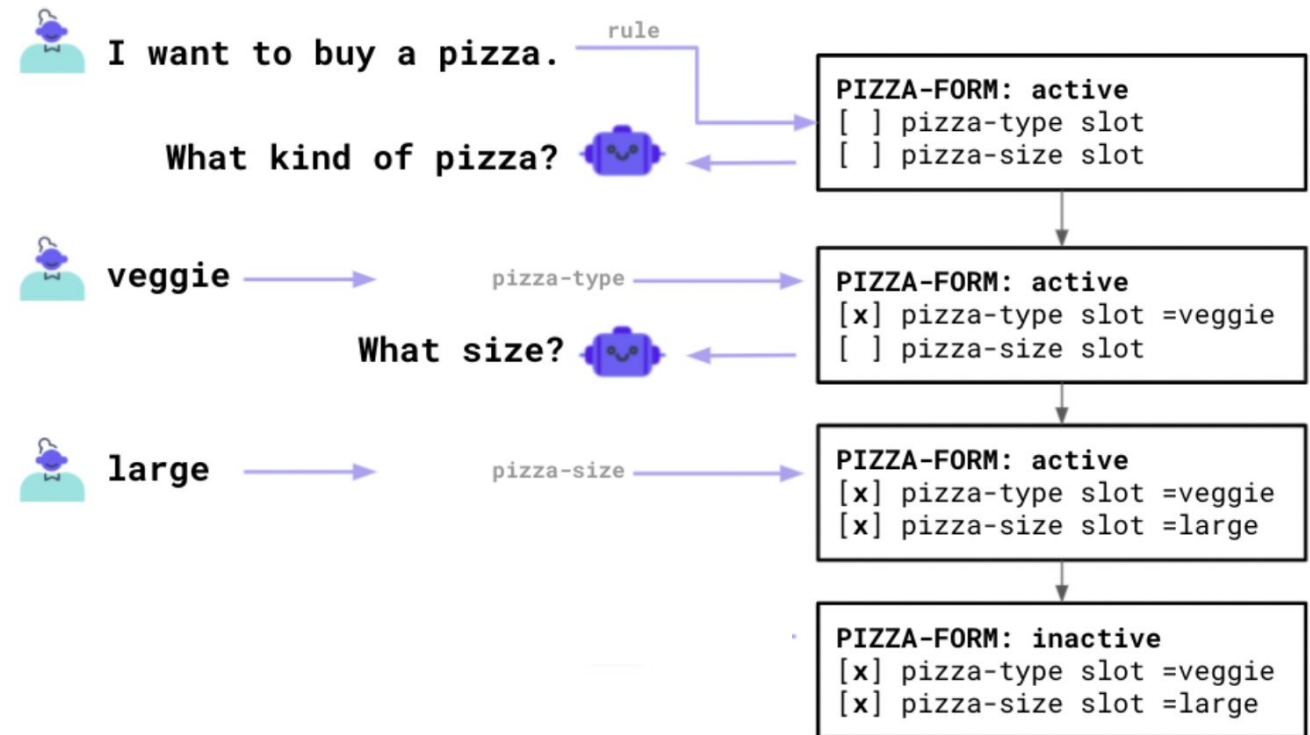
domain file

```
forms:  
  pizza-form:  
    required_slots:  
      - pizza-type
```

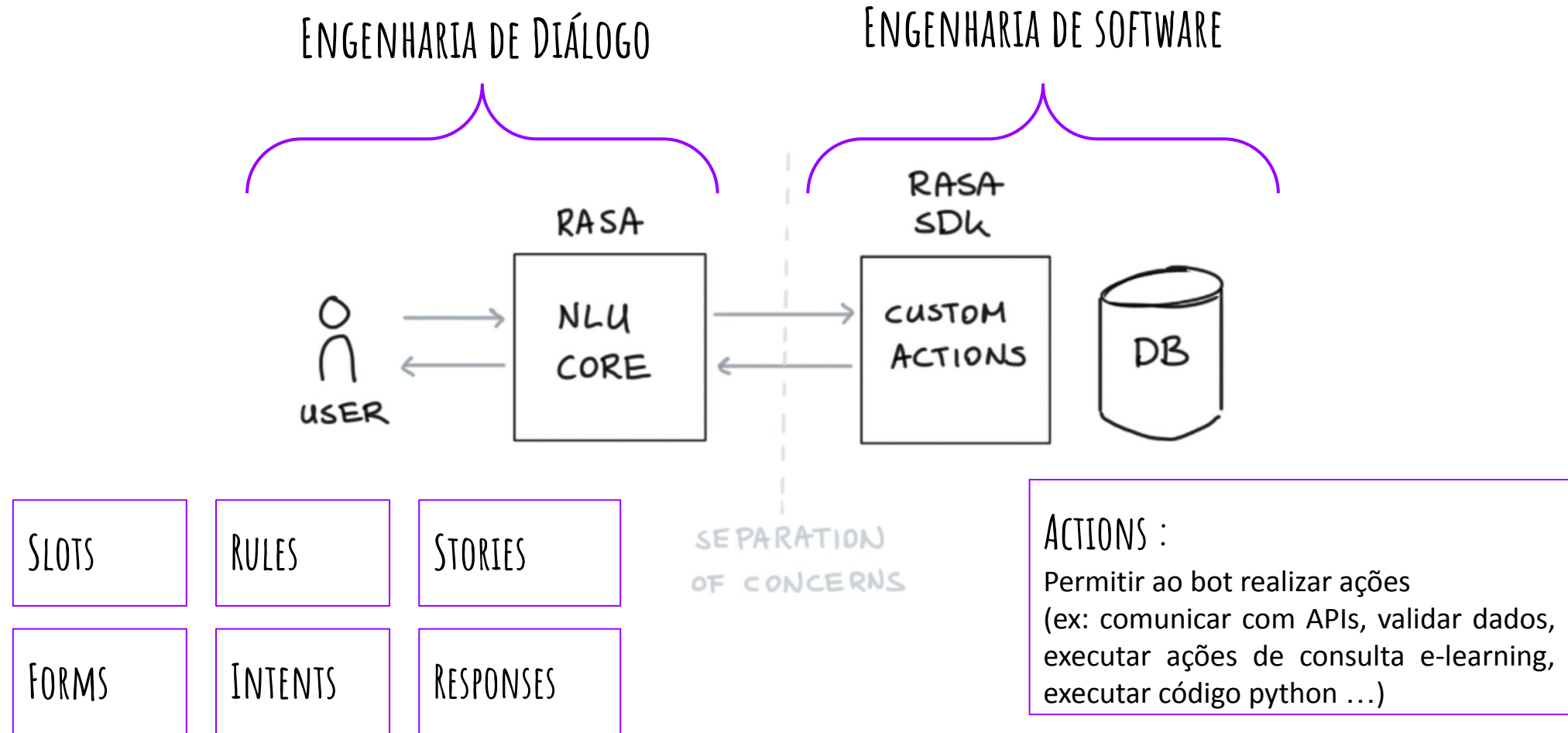
- rule: Activate Pizza Form
steps:
 - intent: buy_pizza
 - action: simple_pizza_form
 - active_loop: simple_pizza_form
- rule: Submit Pizza Form
Ensure that we're starting from an active loop
condition:
 - active_loop: simple_pizza_formsteps:
 - action: simple_pizza_form
 - # If there are no more requested slots*
 - active_loop: null
 - slot_was_set:
 - requested_slot: null
 - # Then submit the form*
 - action: utter_submit
 - action: utter_pizza_slots

Slots: Memória de longa duração

Forms: Ciclo que executa e pergunta informação ao user até que todos os slots associados estejam preenchidos corretamente;



Áreas de trabalho



[Fonte](#)

Alguns conceitos

Intent:

Classifica o input do utilizador humano.

Story:

Define os caminhos que a conversa pode seguir.

Entity:

Permite guardar partes de informação retiradas da mensagem do utilizador.

Response:

Define as respostas do bot.


Rules:

Define partes da conversa que seguem sempre o mesmo padrão.

Action:

Permite ao bot realizar ações (ex: comunicar com APIs, validar dados, executar código python ...)

... fornecer exemplos de conversas a partir dos quais o sistema **aprende** e faz **generalizações** de padrões nos dados;



INTRODUCTION TO RASA


REPRODUZIR TUDO

Conversational AI with Rasa Open Source 3.x

14 vídeos • 28.060 visualizações • Última atualização em 7 de dez. de 2021

⋮ ↺ ↻ ⋮

This playlist contains a full course on how to build Conversational AI with Rasa Open Source 3.x.

 **Rasa** **INSCREVER-SE**

Rasa Learning Center

Welcome to the Rasa Learning Center,
the best place to learn all about Conversational AI.

You can learn all about ...

 **Rasa Features**

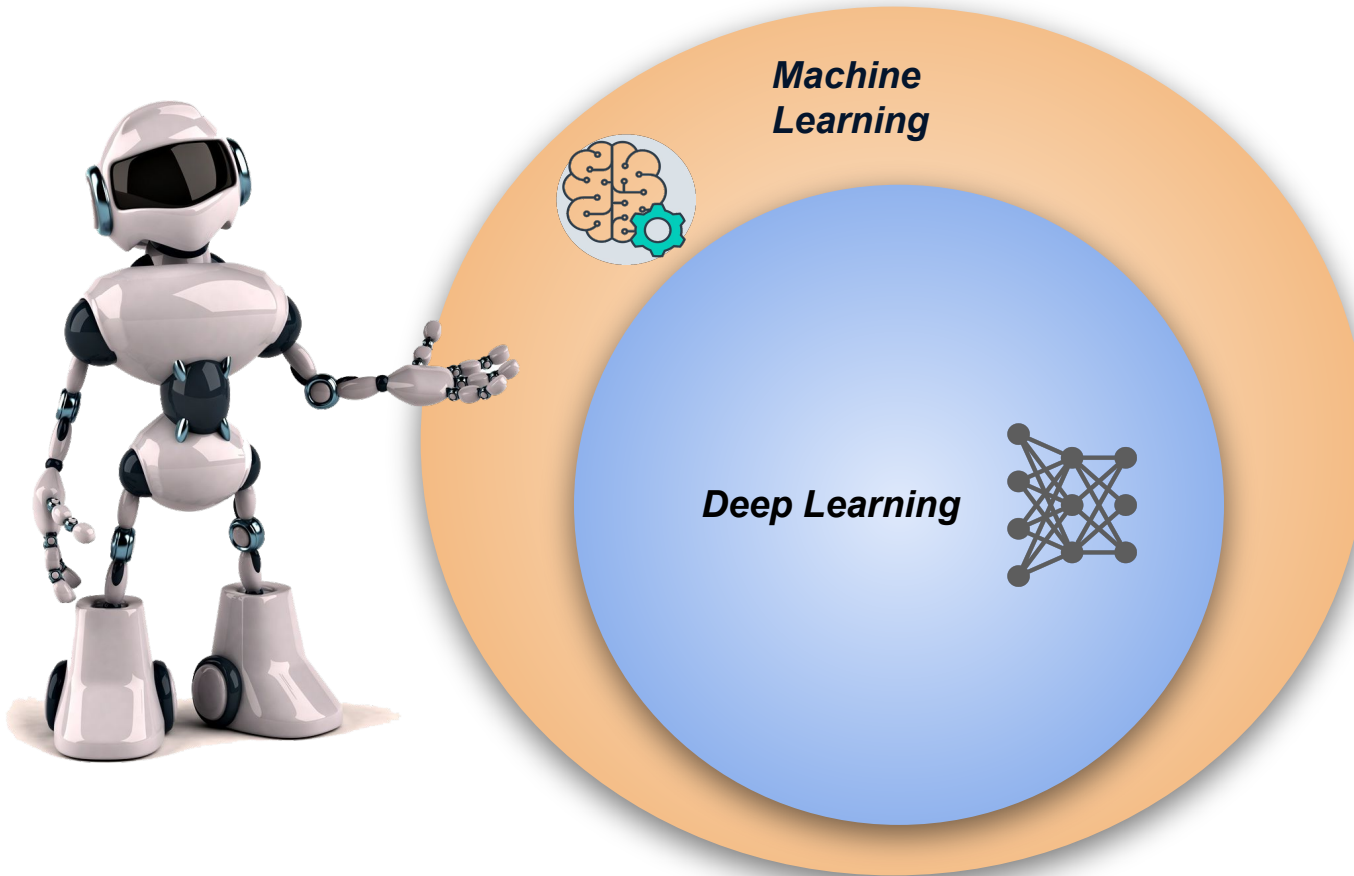
 **Deployment**

 **Conversational Design**

 **Machine Learning**

<https://www.youtube.com/playlist?list=PL75e0qA87dIEjGAc9j9v3a5h1mxl2Z9fi>
<https://learning.rasa.com/>

Deep Learning and DIET classifier



- **Machine Learning:** tipo de inteligência artificial que permite ao computador aprender, de forma a tentar ser mais preciso a prever resultados sem ser programado para tal.
- **Deep Learning:** é uma subárea de machine learning que tem algoritmos inspirados na estrutura e funcionamento do cérebro humano, estes algoritmos chamam-se redes neuronais artificiais.
 - Redes constituídas por nodos (neurónios);
 - Alimentadas com dados de forma a aprenderem;
 - Nodos têm pesos que vão sendo atualizados conforme treinados;
 - Objetivo: obter uma classificação com o máximo de precisão.

Deep Learning and DIET classifier

Big Data and Intelligence Hello



- Base de dados com um número muito elevado de dados;
- Dados tendem a ser compostos por vários campos;
- A informação tende a ser atualizada várias vezes ao longo do tempo.

Big Data and Intelligence Hello

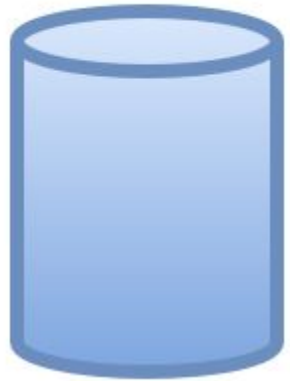


As 5 principais características de Big Data:

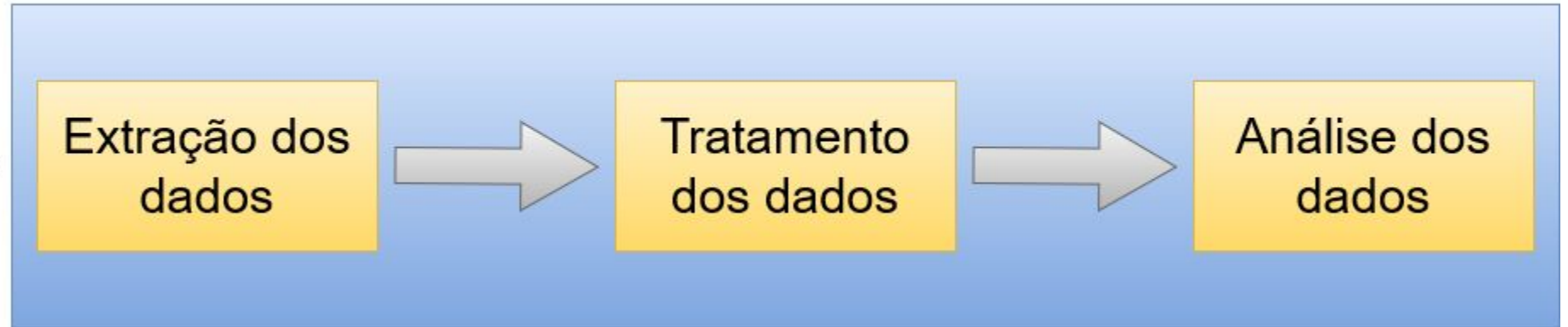
- **Volume** - A quantidade de dados disponíveis;
- **Velocidade** - A velocidade a que os dados são gerados;
- **Variedade** - A variedade dos tipos de dados;
- **Veracidade** - O nível de confiança dos dados;
- **Valor** - O valor da informação que os dados podem dar;

Big Data and Intelligence Hello

E-Schooling



Hello



Três fases na utilização da tecnologia **Big Data** no contexto da plataforma **Hello**:

- Extração
- Processamento
- Análise e conclusão

Exercício 1

Crie um chatbot capaz de:

1. Realizar ações básicas;
2. Conectar-se com o módulo de **Education Intelligence**:
 - Utilizar o número interno do aluno (que poderá ser facultado por ele na interação, a pedido do bot) para identificar possíveis situações de risco (ex: as notas encontram-se separadas por mais do que dois valores).
3. Conectar-se com módulos externos (código python):
 - **Análise de sentimentos**;
4. Conectar-se a API cliente (**simulador e-schooling**)

Possível solução

Exercício 1: Overview arquitetura

Actions server

`rasa run actions`

Rasa

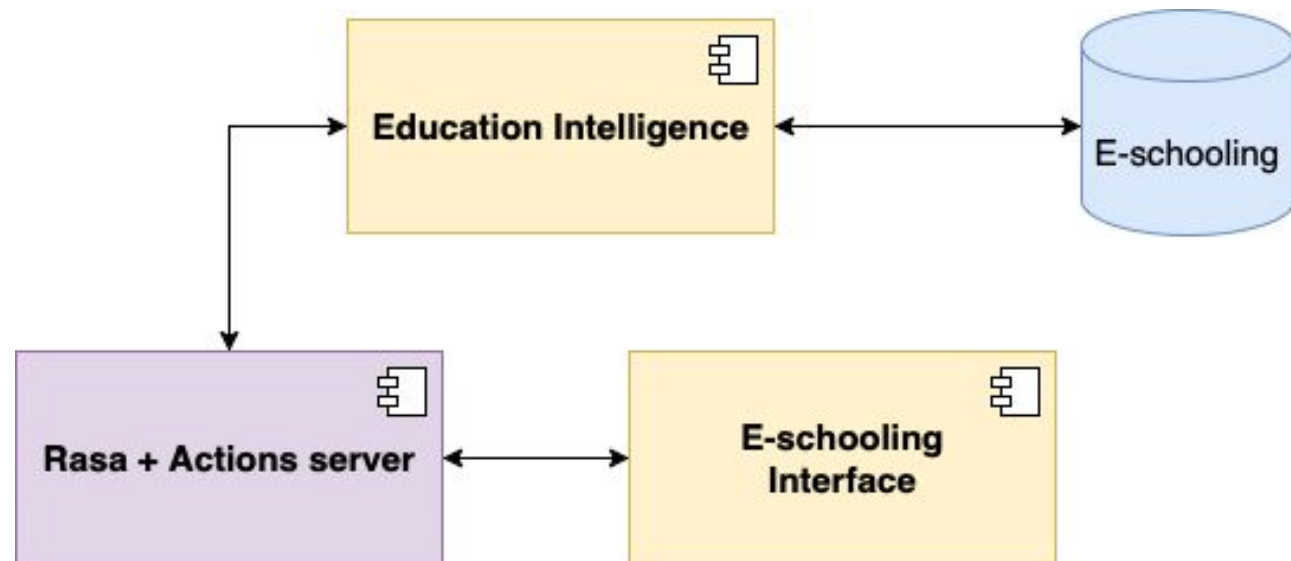
`rasa run --enable-api --cors="*"`

E-schooling Interface

1. `python -m http.server`
2. Aceder ao file bot.html (simula e-schooling interface)

Simulador Education Intelligence (servidor json-server)

`npm run json-server --watch notas.json`

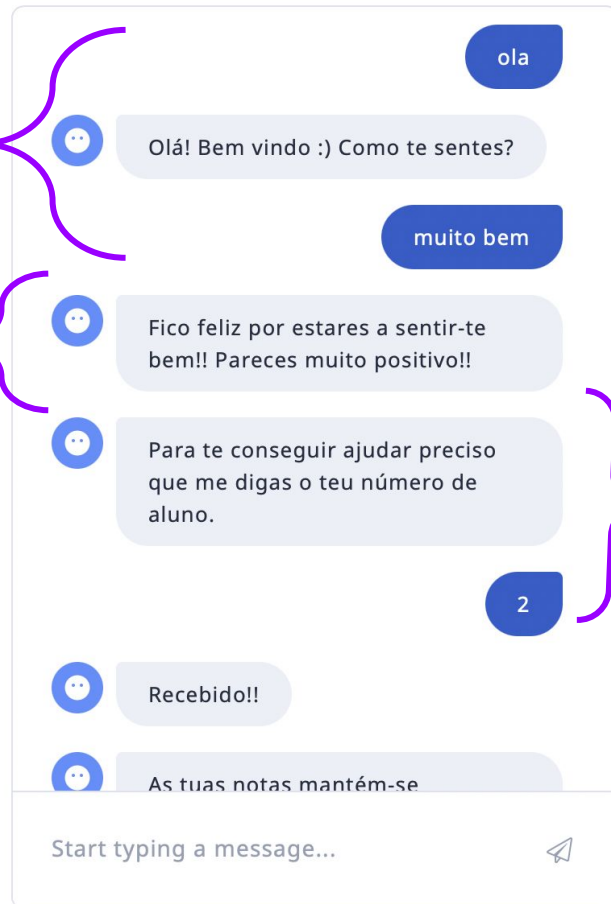


Possível solução

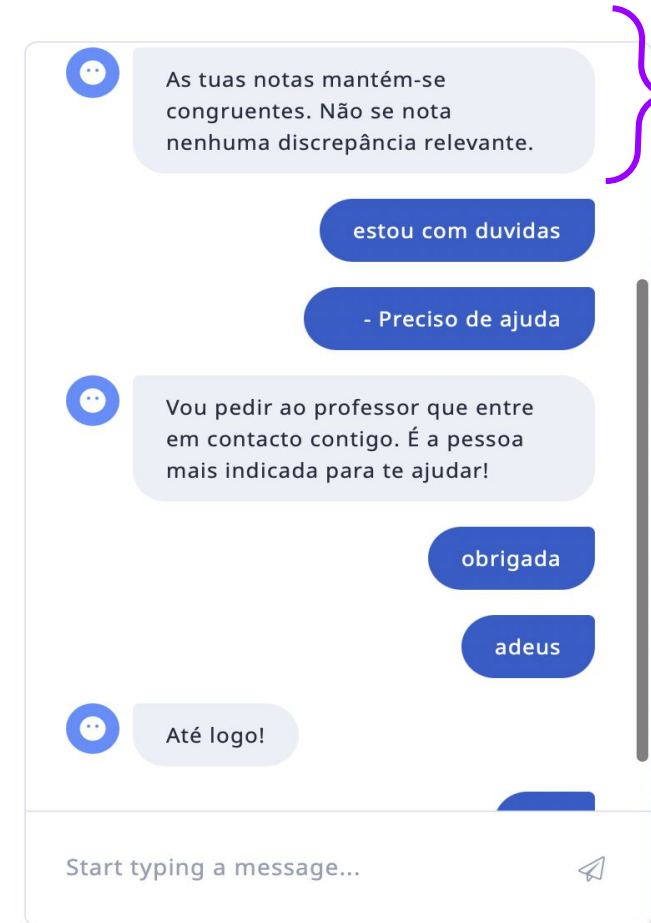
Exercício 1: Exemplo de interação

Cumprimentar
o aluno

Conexão módulos
externos.
Resultado **análise
sentimentos**;



Formulário é
ativado;
(student
_number)



Conexão **Education
Intelligence**:
Procura por situação
de risco

Possível solução

Instalação

- 1) Seguir as indicações de acordo com o sistema operativo:

<https://rasa.com/docs/rasa/installation/>

- 2) Criar um novo projeto:

```
rasa init
```

Quick Installation

Isolate your python project using a virtual environment.

• Ubuntu • macOS • Windows

Create a new virtual environment by choosing a Python interpreter and making a `./venv` directory to hold it:

```
python3 -m venv ./venv
```

Activate the virtual environment:

```
source ./venv/bin/activate
```

Install Rasa Open Source using pip (requires Python 3.7, or 3.8).

```
pip3 install -U --user pip && pip3 install rasa
```

```
Welcome to Rasa! 🤖
```

```
To get started quickly, an initial project will be created.  
If you need some help, check out the documentation at https://rasa.com/docs/rasa.  
Now let's start! 🗨️
```

```
? Please enter a path where the project will be created [default: current directory] █
```

Exercício 2

Crie um chatbot que ajude os aluno a preencher um formulário para contacto com os docente:

- Cumprimentar o aluno;
- Questionar qual o professor que deseja contactar;
- Solicitar um e-mail para o professor entrar em contacto;
- Informar o aluno que o professor já foi informado;
- Despedir-se;

```
[Your input ->  ola  
Ola! Em que posso ajudar?  
[Your input ->  Queria falar com um professor. É possível?  
Que professor quer contactar?  
[Your input ->  Vitor Neves  
Qual é o seu email?  
[Your input ->  maria.a.b2000@hotmail.com  
O teu pedido foi enviado ao docente.  
O professor Vitor Neves vai entrar em contacto através do email maria.a.b2000@ho  
tmail.com!  
[Your input ->  _obrigada
```

comandos:

```
> rasa train  
> rasa shell
```

Exercício 2

ola

Ola! Em que posso ajudar?

Quero falar com um professor. É possível?

Que professor quer contactar?

Pedro Pires

Qual é o seu email?

maria.a.b2000@hotmail.com

O teu pedido foi enviado ao docente.

O teu pedido foi enviado ao docente.

O professor Pedro Pires vai entrar em contacto através do email maria.a.b2000@hotmail.com!

Obrigada

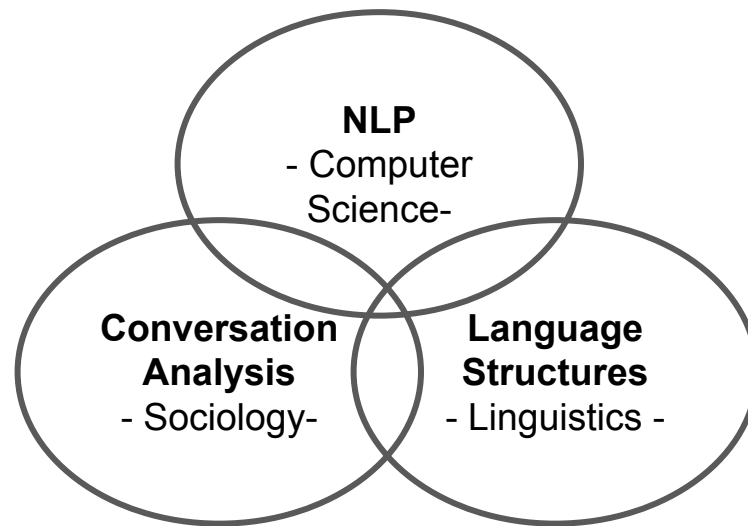
Adeus

Start typing a message...

Conversational UX

"As técnicas de Processamento de Língua Natural permitem a análise automática de partes da língua. No entanto, a junção dessas partes em estruturas que ocorrem em uma conversa é responsabilidade do projetista de UX."

Dialogue Design



Estilos

1. *System Centric*: CLI com NLU

- within-turn context
- two-turn interaction model
- recognizes commands or queries
- user-driven voice or visual



2. *Content Centric*: FAQ com NLU

- within-turn context
- recognizes questions
- document-like, topic-based responses
- user-driven



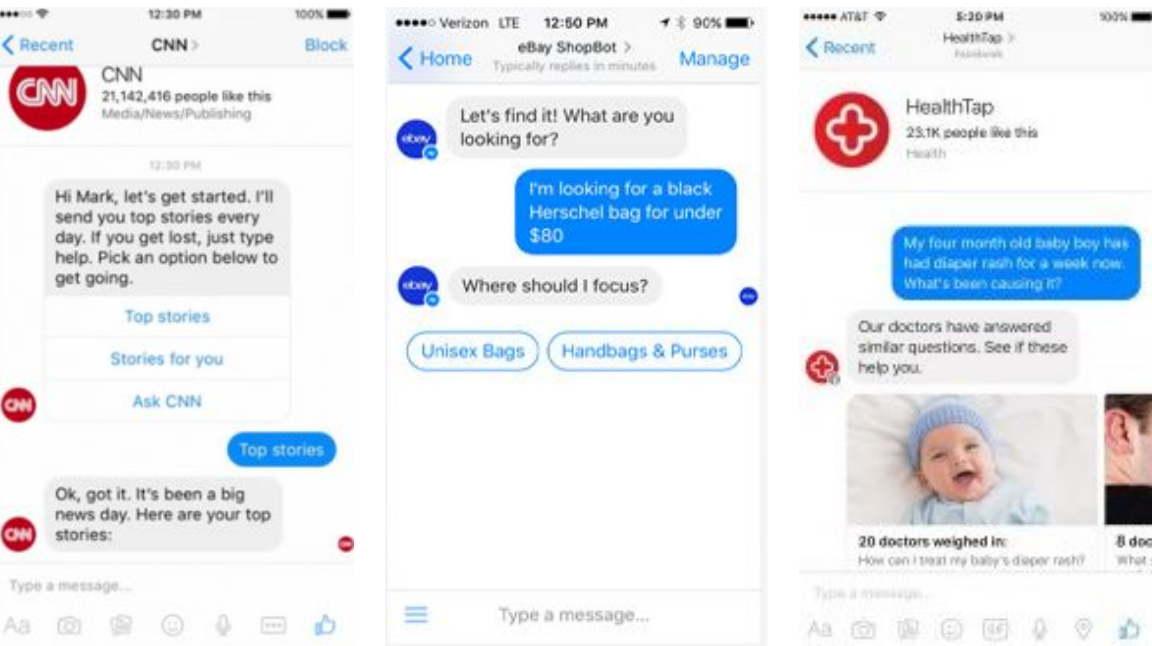
3. *Visual Centric*: Botões em meio ao diálogo

- within-turn context
- requires direct manipulation
- buttons, menus, lists, images, etc.
- agent-driven

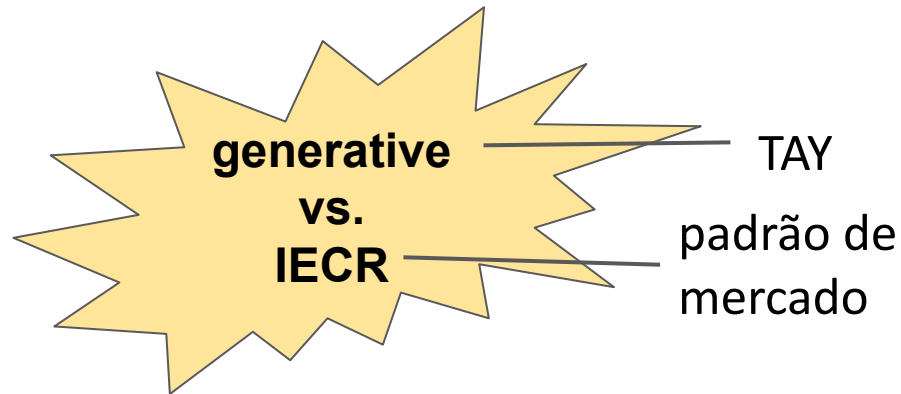


4. *Conversation First*:

- like natural conversation
- cross-turn
- persistent context
- recognizes conversational actions
- fine-grained back-and-forth
- mixed initiative

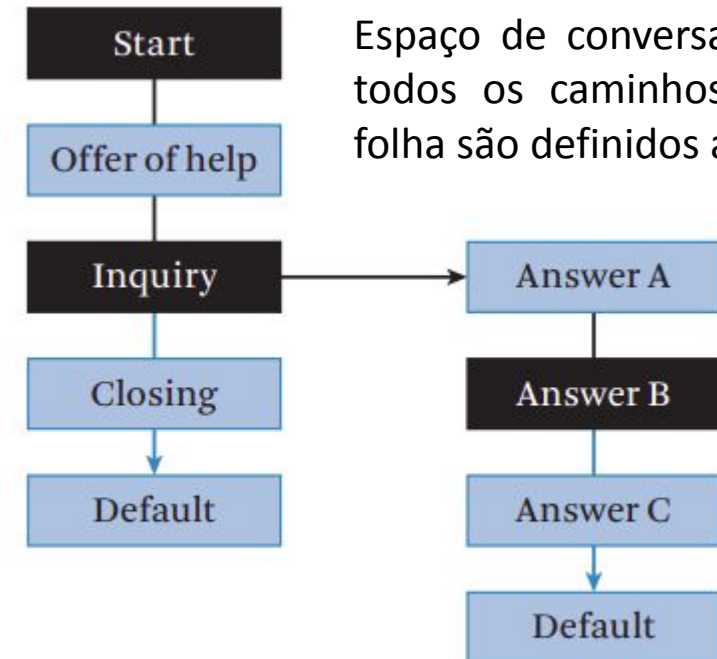


Intent-Entity-Context-Response (IECR)



IECR Pipeline

1. identificação de intents or classes de frase por machine learning
2. extração de entidades orientada por palavras chave
3. variáveis de contexto (*e.g.* nome)
4. respostas textuais



Espaço de conversa finito, i.e. todos os caminhos da raiz à folha são definidos a priori.

Dialog tree (branch level).

IECR Design

1. Identifique as classes (ou intents)

Class	Example
#greeting	hello, hi, hey, hi there, yo, ...
#welfare_check	how are you?, how's it going?, how do you do?, ...
#capability_check	what can you do?, what do you know?, ...

2. Identifique as entidades

Class	User	Agent B
#flight_request	I want flights to New York	Here are all flights to New York.
#cheap_flight_request	show me the cheapest flight to New York	July 10-15 is the cheapest time.
#budget_flight_request	I want flights under \$300	Here are some cities under \$300.

3. Estabeleça contexto consequencial talvez o maior desafio de UX

U: Quero marcar um voo para meu filho.

A: Filho fica em qual país?

4. Escreva as respostas

o conhecimento deve ser dinâmico

U: Quero ir na Taberna Belga.

A: Está encerrada hoje.

5. Crie Ramos

Multiple User Entry Points.

```
1 user input A
2 if #flight_request
3   say "Where do you want to go?"
4   user input B
5     if @city
6       set $destination to @city
7       say "Where are you leaving from?"
8       user input C
9         if @city
10           set $origin to @city...
```

Validação do Diálogo

Validação de UX se dá por

- **Inspeção** - exame dos aspectos de UX
- **Testes** - validação por utilizadores alvo
- **Inquérito** - utilização de grupos focais

TABLE VII. SAMPLE SIZE ESTIMATION FOR VARIOUS USER TESTING PURPOSES

Main Purpose	# users
To find more cosmetic problems and problems relating to structure and content.	5
To find few major and more minor problems. Also, this is more appropriate for commercial studies and more problems in layout and formatting.	8
To find more catastrophic, major, minor and cosmetic problems; also, for finding more problems relating to design, navigation and the key aims and functions for which the system is built. Moreover, it is more appropriate for comparative studies.	16±4
For statistically significant studies and analysis of the performance metrics, such as success rate	≥ 20

Alroobaea, R., & Mayhew, P. J. (2014). *How many participants are really enough for usability studies?* 2014 Science and Information Conference. doi:10.1109/sai.2014.6918171

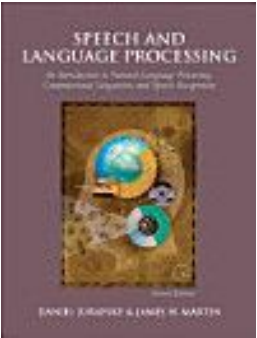
Chapter 3 Safety First: Conversational Agents for Health Care



Timothy Bickmore, Ha Trinh, Reza Asadi and Stefan Olafsson

Abstract Automated dialogue systems represent a promising approach for health care promotion, thanks to their ability to emulate the experience of face-to-face interactions between health providers and patients and the growing ubiquity of home-based and mobile conversational assistants such as Apple’s Siri and Amazon’s Alexa. However, patient-facing conversational interfaces also have the potential to cause significant harm if they are not properly designed. In this chapter, we first review work on patient-facing conversational interfaces in healthcare, focusing on systems that use embodied conversational agents as their user interface modality. We then systematically review the kinds of errors that can occur if these interfaces are not properly constrained and the kinds of safety issues these can cause. We close by outlining design recommendations for avoiding these issues.

Bickmore, T., Trinh, H., Asadi, R., & Olafsson, S. (2018). Safety first: conversational agents for health care. In *Studies in Conversational UX Design* (pp. 33-57). Springer, Cham.

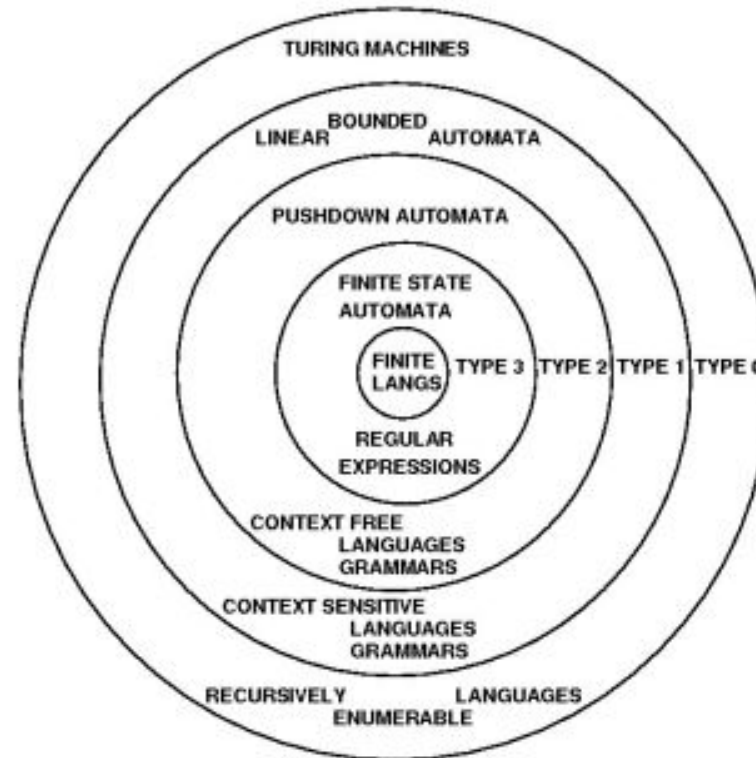


Chatbots and Digital Assistants

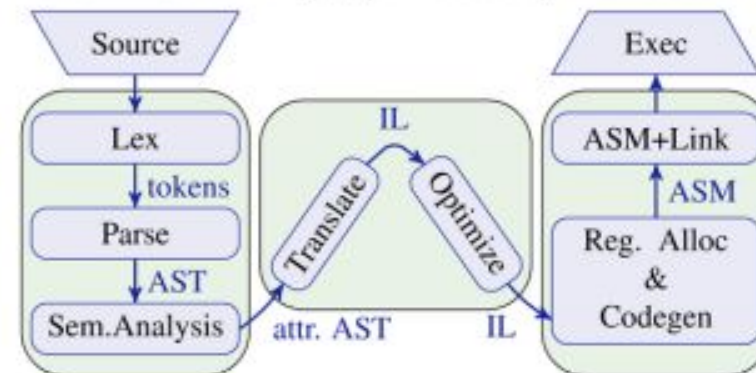
Tutorial sobre RASA Hello Project

19/09/2022

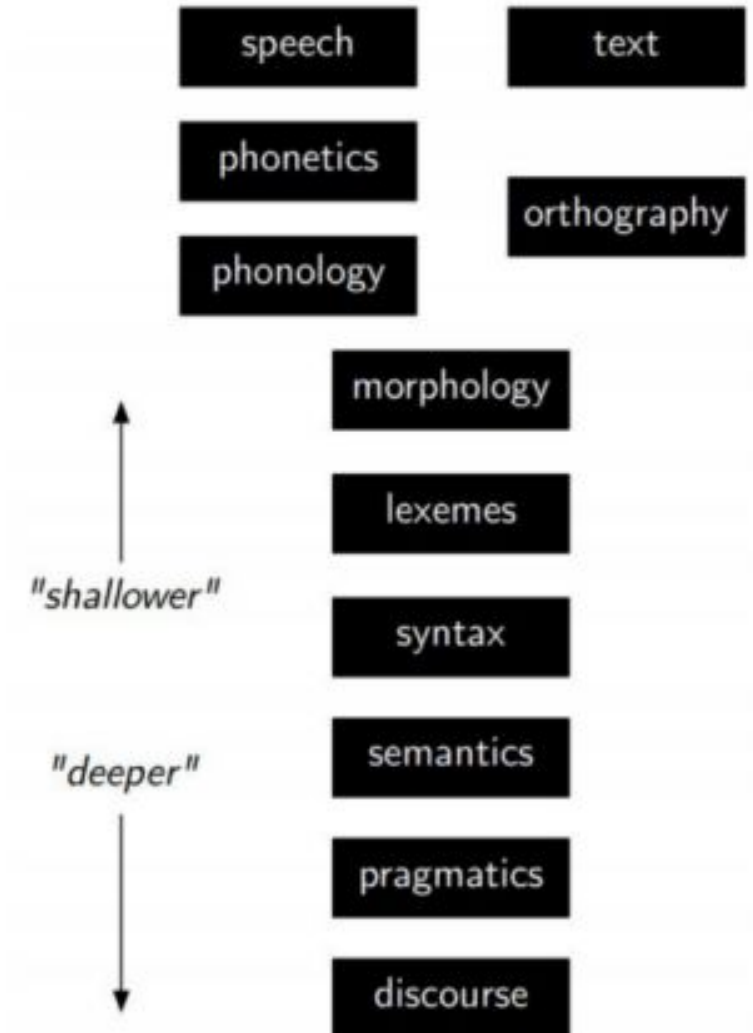
Anexo I - Linguagens Formais



Fonte: Natural Language Processing: Introduction to NLP



Fonte: Describe local and global optimization ?



Fonte: Formal Languages and Automata theory

Anexo II - Língua Natural

