

ChatBOT

Tópicos de Inteligência Artificial

Prof. Dr. [Dieval Guizelini](#)

Março/2023

Antes de começar

- Finalidade
- Quais as P&R (FAQ) que estão disponíveis sobre o tema
- A forma mais simples de um chatbot é construir um fluxo de opções (árvores de perguntas e respostas)
- Pode-se construir um mapa mental de intenções (verbos)
- Alguns chatbots famosos:
 - Eliza (1966)
 - Parry (1972)
 - A.L.I.C.E. (1995)
 - Smarter Child (2001)
 - SIRI (2010)
 - Google now (2012)
 - Alexa (2015)

Tipos e abordagens

Tipos:

- Baseados em texto
 - Baseados em voz
-
- Regras de associação
 - Redes neurais ou NLP

Os modelos de Processamento de Linguagem Natural (NLP) mais recentes/promissores

- BERT (2018): Pre-training of Deep Bidirectional Transformers for Language Understanding
modelo baseado na implementação do “multi-layer bidirectional Transformer encoder” no Tensor2tensor
<https://arxiv.org/pdf/1810.04805.pdf>

BERT realiza bem uma ampla variedade de tarefas. Obteve resultados surpreendentes nos benchmarks. No SQuAD v1.1, o BERT alcançou 93,2% na nota F1, que mede a precisão. No GLUE, o alcance foi de 7,6%.

- T5 (2019) - Text-To-Text Transfer (Google): Exploring the Limits of Transfer Learning with a Unified Text-toText Transformer

É um modelo de rede neural com 11 bilhões de parâmetros que converte tarefas de linguagem em um formato texto para texto. Após ajuste fino, este modelo alcançou estado da arte em mais de 20 tarefas, em benchmarks como GLUE, SuperGLUE, SQuAD e CNN/Daily Mail.

Diferentemente dos modelos GPT-2 e GPT-3, que são treinados para prever qual a próxima palavra a partir de um input, a geração de texto do T5 acontece através do método de preenchimento do espaço vazio.

<https://arxiv.org/abs/1910.10683>



GPT-2, GPT-3, GPT-3.5... ChatGPT

o GPT-3 é um modelo de 175 bilhões de parâmetros – 100x maior que a versão do GPT-2, considerado **um modelo gerador de texto**. O público foi apresentado ao GPT-3 por meio do artigo **Language Models are Few-Shot Learners**, que apresenta o modelo não supervisionado.

<https://arxiv.org/pdf/2005.14165.pdf>

A base de treinamento utilizada foram as discussões do Reddit, corpus do Wikipedia e textos científicos. O GPT-3 foi capaz de responder tarefas de tradução, respostas para perguntas padronizadas e geração de novos textos segundo um estilo sem ajuste fino.

O jornal The Guardian publicou um artigo escrito pelo GPT-3. Partindo da instrução: "Escreva um breve artigo em torno de 500 palavras. Mantenha a linguagem simples e concisa. Foque no porquê os humanos não têm que temer da IA". O GPT-3 gerou oito ensaios que, depois, foram compilados e editados neste artigo.

<https://www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3>

Apesar do sucesso, o GPT-3 não teve bom desempenho no SuperGLUE, apresentando dificuldades na análise de contexto de palavras nos testes WiC e RACE, mas obteve desempenho do estado da arte na compreensão de leitura (COPA, ReCoRD). E ainda, houveram casos em que o modelo gerou conteúdos preconceituosos e enviesados.

Animação com a ideia do modelo GPT-3: <https://jalammar.github.io/how-gpt3-works-visualizations-animations/>



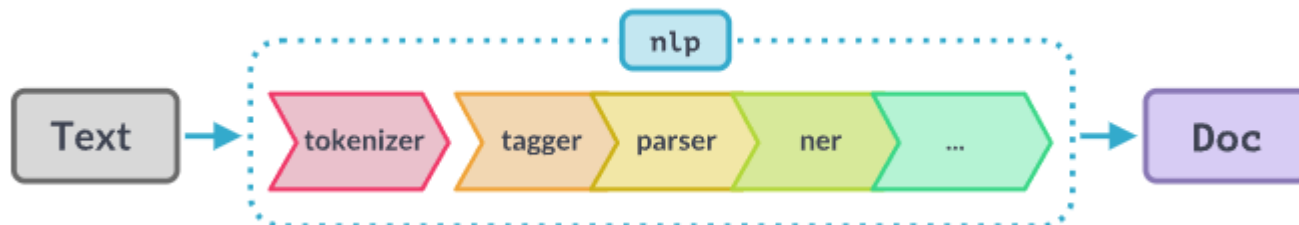
Visão Geral das soluções

Software	Indicado para	Características	Custo/Obs (US)
ProProfs ChatBot	agendar compromissos e automatizar o suporte ao cliente	<ul style="list-style-type: none">- Lógica de ramificação- Fluxos de trabalho personalizados- Transfere chats para os departamentos certos.- Construtor de chatbot de arrastar e soltar.	15 dias de avaliação
Mitsuku – Pandorabot	publicidade, e-learning, assistência virtual, entretenimento e educação.	<ul style="list-style-type: none">- Redução simbólica- Ciclo de segmentação para personalidade de bot refinada- Retenção de registro de bate-papo- API do aplicativo	Free Compartilhado 75 Dedicado 1500/mês
Botsify	Ambientes corporativos e educacionais	<ul style="list-style-type: none">- Formulários de conversação- Árvore de histórias- Integrar com um site, Facebook, Amazon e Slack; Chatbots de educação	
MobileMonkey	Conectar clientes pelas redes sociais.	<ul style="list-style-type: none">- Integração com Facebook Messenger, Zapier, SMS e WebChat; API MobileMonkey; SMS...	

spaCy

- <https://spacy.io/usage>
- Teste o Matcher baseado em regras do spaCy criando padrões de tokens interativamente e executando-os em seu texto. Cada token pode definir vários atributos, como valor de texto, tag de parte da fala ou sinalizadores booleanos.
A visualização baseada em token permite explorar como o spaCy processa seu texto – e por que seu padrão corresponde ou não. (<https://explosion.ai/demos/matcher>)
- Sobre o modelo baseado em regras:
<https://spacy.io/usage/rule-based-matching>

```
pip install -U pip setuptools wheel  
pip install -U spacy  
python -m spacy download pt_core_news_sm
```



<https://spacy.io/usage/processing-pipelines>

Exercício

- Desenvolver um chat, que atenda minimamente, esse exemplo:

```
print("ChatBOT rodando...")
```

```
done = False
```

```
while not done:
```

```
    msg = input(">")
```

```
    if msg == 'FIM' or msg == 'quit' or msg == 'sair' :
```

```
        done = True
```

```
        break
```

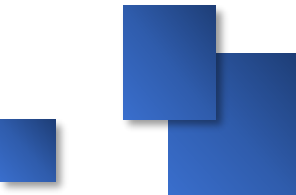
```
    intencao = predicao(msg)
```

```
    res = busca_resposta(intencao, intents)
```

```
    print(res)
```


<https://github.com/NeuralNine/neuralintents>

- Neuralintents é uma das soluções mais simples, as configurações das mensagens são realizadas no formato json.



Get input
Get input from some source
(console, API, speech recognition, etc.)



Process input
The input statement is processed by each of the logic adapters.

Logic adapter 1

1. Select a known statement that most closely matches the input statement.
2. Return a known response to the selected match and a confidence value based on the matching.

Logic adapter 2

1. Select a known statement that most closely matches the input statement.
2. Return a known response to the selected match and a confidence value based on the matching.

Return the response from the logic adapter that generated the highest confidence value for its result.



Return response
Return the response to the input
(console, API, speech synthesis, etc.)



```
chatbot = ChatBot(  
    "My ChatterBot",  
    logic_adapters=[  
        {  
            "import_path": "chatterbot.logic.BestMatch",  
            "statement_comparison_function": chatterbot.comparisons.LevenshteinDistance,  
            "response_selection_method": chatterbot.response_selection.get_first_response  
        }  
    ]  
)
```

ChatterBot

- TimeLogicAdapte

User: What time is it?

Bot: The current time is 4:45PM.

- MathematicalEvaluation

User: What is four plus four?

Bot: $(4 + 4) = 8$

Specific response example

```
from chatterbot import ChatBot











# Create a new instance of a ChatBot
bot = ChatBot(
    'Exact Response Example Bot',
    storage_adapter='chatterbot.storage.SQLStorageAdapter',
    logic_adapters=[
        {
            'import_path': 'chatterbot.logic.BestMatch'
        },
        {
            'import_path': 'chatterbot.logic.SpecificResponseAdapter',
            'input_text': 'Help me!',
            'output_text': 'Ok, here is a link: http://chatterbot.rtfld.org'
        }
    ]
)

# Get a response given the specific input
response = bot.get_response('Help me!')
print(response)
```

ChatterBot “Corpus”

```
1  categories:
2  - cumprimentos
3  conversations:
4  - - Olá
5    - Oi
6  - - Oi
7    - Olá
8  - - Saudações!
9    - Olá
10 - - Olá
11   - Saudações!
12 - - E aí como vai?
13   - Bem
44 - - E aí beleza?
45   - Não muito bem.
46 - - E aí beleza?
47   - Não muito bem.
48 - - E aí beleza?
49   - Não muito bem, e você?
50 - - E aí beleza?
51   - Nada de mais.
52 - - E aí beleza?
53   - O céu esta acima, mas eu estou bem, obrigado. E você?
54 - - E aí suave?
55   - Suavidade!
56 - - E aí brother?
57   - E aí bro!
58 - - Tudo bom?
59   - Melhor agora
60 - - E aí cara?
61   - Tudo bom?
62   - Tudo de bom e você?
```

ChatterBot “Corpus”

1	categories:	44	- - E aí beleza?		
2	- cumprimentos	45	- Não muito bem.		
3	conversations:	46	- - E aí beleza?		
4	- - Olá	47	- Não muito bem.		
5	- Oi	48	- - E a	..	
6	- - Oi	49	- Não		
7	- Olá	50	- - E a	 compliment.yml	Support multiple category tags for a corpus
8	- - Saudações!	51	- Nad	 conversations.yml	Add answer to the fundamental question
9	- Olá	52	- - E a		
10	- - Olá	53	- O c	 games.yml	Added 1 new question with multiple answers
11	- Saudações!	54	- - E a	 greetings.yml	Added 6 popular brazilian greetings
12	- - E aí como vai?	55	- Sua		
13	- Bem	56	- - E a	 linguistic_knowledge.yml	Remove lines with invalid length
		57	- E a	 money.yml	Fix invalid characters and lines
		58	- - Tud		
		59	- Me1	 proverbs.yml	Add 3 proverbs - lines 67-72
		60	- - E a	 suggestions.yml	Create games.yml and money.yml
		61	- Tud		
		62	- Tud	 trivia.yml	Add informations about Brazil.
				 unilab.yml	Support multiple category tags for a corpus

Distância de Levenshtein

- Quantas “operações” são necessárias para transformar str1 em str2?

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

if $\min(i, j) = 0$,

otherwise.

		0	1	2	3	4	5	6	7	(index)
		#	S	I	T	T	I	N	G	
0	#	0	1	2	3	4	5	6	7	
1	K	1								
2	I	2								
3	T	3								
4	T	4								
5	E	5								
6	N	6								

(index)

		0	1	2	3	4	5	6	7	(index)
		#	S	I	T	T	I	N	G	
0	#	0	1	2	3	4	5	6	7	
1	K	1	1	2	3	4	5	6	7	
2	I	2	2	1	2	3	4	5	6	
3	T	3	3	2	1	2	3	4	5	
4	T	4	4	3	2	1	2	3	4	
5	E	5	5	4	3	2	2	3	4	
6	N	6	6	5	4	3	3	2	3	

(index)

Em python

```
import numpy as np

def levenshtein(seq1, seq2):
    size_x = len(seq1) + 1
    size_y = len(seq2) + 1
    matrix = np.zeros ((size_x, size_y))
    for x in xrange(size_x):
        matrix [x, 0] = x
    for y in xrange(size_y):
        matrix [0, y] = y

    for x in xrange(1, size_x):
        for y in xrange(1, size_y):
            if seq1[x-1] == seq2[y-1]:
                matrix [x,y] = min(
                    matrix[x-1, y] + 1,
                    matrix[x-1, y-1],
                    matrix[x, y-1] + 1
                )
            else:
                matrix [x,y] = min(
                    matrix[x-1,y] + 1,
                    matrix[x-1,y-1] + 1,
                    matrix[x,y-1] + 1
                )
    print (matrix)
    return (matrix[size_x - 1, size_y - 1])
```

Ferramentas

- CHatGPT da openAI
<https://chat.openai.com/>
- ChatBot com o IBM Watson
- Dialogflow (Google)
- Soluções em Python
<https://github.com/NeuralNine/neuralintents>

Referencias e conjunto de dados

- PLOS COMPUTER & INFORMATION SCIENCES. Text Mining:Curated Collections, 2016
- Mikolov et al, Efficient Estimation of Word Representations in Vector Space, 2013
- Gensim word2vec
<https://radimrehurek.com/gensim/>
- Tensorflow
<https://www.tensorflow.org/tutorials/text/word2vec>
- Chatterbot
<https://chatterbot.readthedocs.io/en/stable/index.html>
- SpaCy
<https://spacy.io/>

