Processamento de Linguagem Natural

Prof. Henrique Batista da Silva

PUC Minas Virtual

Prof. Henrique Batista da Silva

- Graduado em Sistemas de Informação - PUC Minas (2008).
- Mestre em Informática pela PUC Minas - Microsoft Innovation Center (2011).

Prof. Henrique Batista da Silva

- Atuação profissional: Cientista de Dados
- Área de atuação: visão computacional, aprendizado de máquina/deep learning, banco de dados.
- Ensino: graduação e pós (lato sensu) desde 2010.

Apresentação da disciplina

PUC Minas Virtual

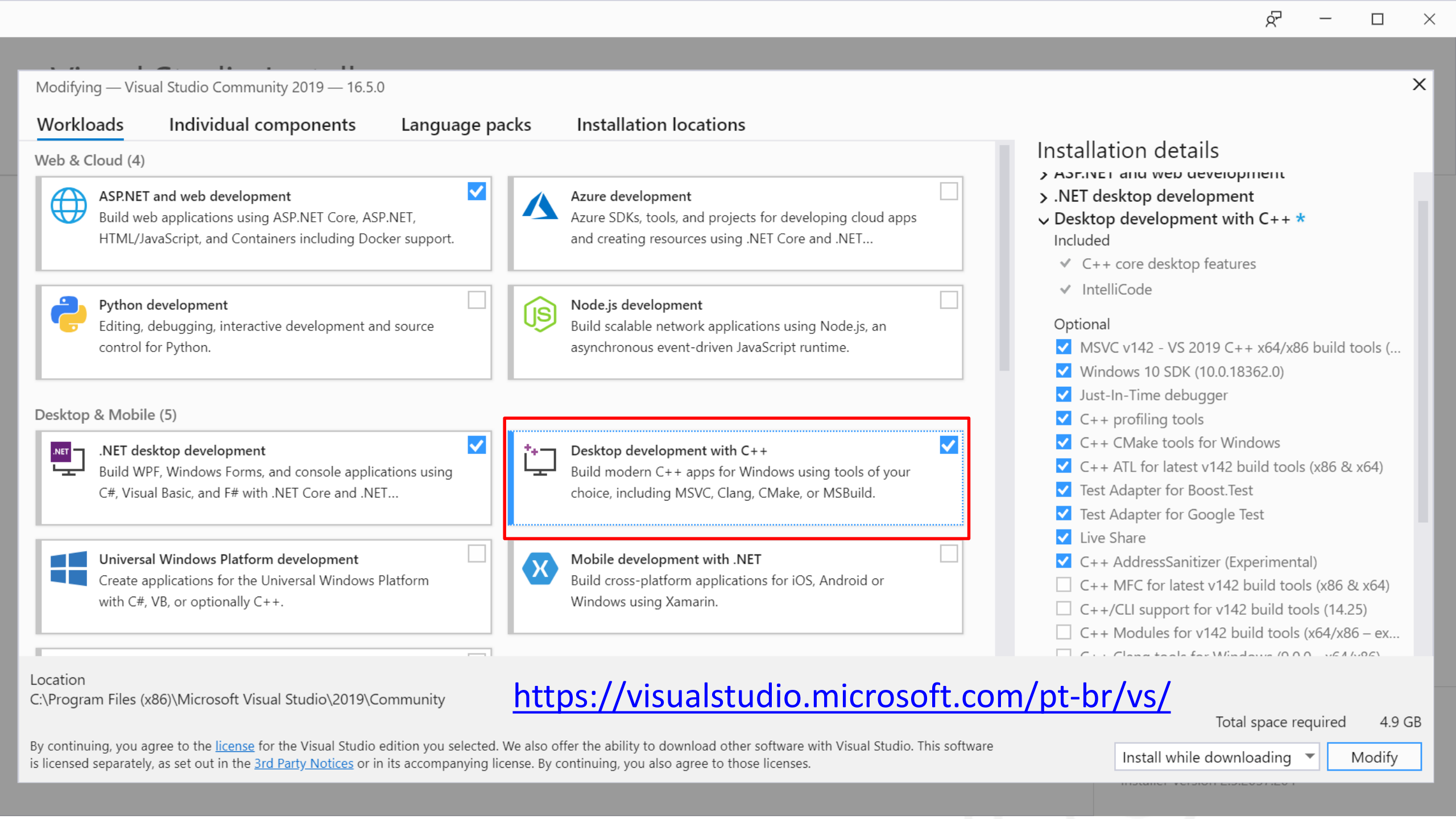
Tópicos de estudo

- Algoritmos e técnicas de processamento em linguagem natural.
- Expressões regulares. Medidas de similaridade textual (TF-IDF). Parsing, tokenização, lematização, stemming.
- Extração de informação. Arquitetura de aplicação para processamento de Linguagem Natural.
- Análise de sentimento.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Ferramentas

- Python 3 (os testes foram realizados no 3.7.7)
- numpy-1.18.2
- pandas-1.0.3
- nltk-3.4.5
- scikit_learn-0.22.2
- vaderSentiment
- Visual Studio Community C++ (apenas para Windows)
- nlpia-0.2.14


Modifying — Visual Studio Community 2019 — 16.5.0
Workloads
Individual components
Language packs
Installation locations
Web & Cloud (4)
ASP.NET and web development
Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker support.
Azure development
Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET Core and .NET...
Python development
Editing, debugging, interactive development and source control for Python.
Node.js development
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.
Desktop & Mobile (5)
.NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET Core and .NET...
Desktop development with C++
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.
Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.
Mobile development with .NET
Build cross-platform applications for iOS, Android or Windows using Xamarin.
Installation details
.NET desktop development
Desktop development with C++ *
Included
C++ core desktop features
IntelliCode
Optional
MSVC v142 - VS 2019 C++ x64/x86 build tools (...
Windows 10 SDK (10.0.18362.0)
Just-In-Time debugger
C++ profiling tools
C++ CMake tools for Windows
C++ ATL for latest v142 build tools (x86 & x64)
Test Adapter for Boost.Test
Test Adapter for Google Test
Live Share
C++ AddressSanitizer (Experimental)
C++ MFC for latest v142 build tools (x86 & x64)
C++/CLI support for v142 build tools (14.25)
C++ Modules for v142 build tools (x64/x86 – ex...
Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community
https://visualstudio.microsoft.com/pt-br/vs/
Total space required 4.9 GB
By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.
Install while downloading
Modify

Introdução

Linguagem natural vs. linguagem de programação

- Linguagens naturais são diferentes de linguagem de programação.
  - Linguagens naturais são o que os humanos usam para compartilhar informações entre si
  - Não utilizamos compiladores para “traduzir” o que um ser humano tem a dizer para o outro.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Linguagem natural vs. linguagem de programação

- Definição (Processamento de Linguagem Natural):
  - Área de pesquisa em ciência da computação e inteligência artificial (IA) relacionada ao processamento de idiomas naturais (português, inglês, etc.).
  - Geralmente envolve a tradução de linguagem natural em dados que um computador possa "entender".

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Linguagem natural vs. linguagem de programação

- Seria intuitivo pensar em um programa interpretador com um interpretador Python e faça a leitura e toma uma determinada ação.
- No entanto, esta ações não são definidas com precisão. Existe um pipeline de processamento de linguagem natural que deverá ser criado.

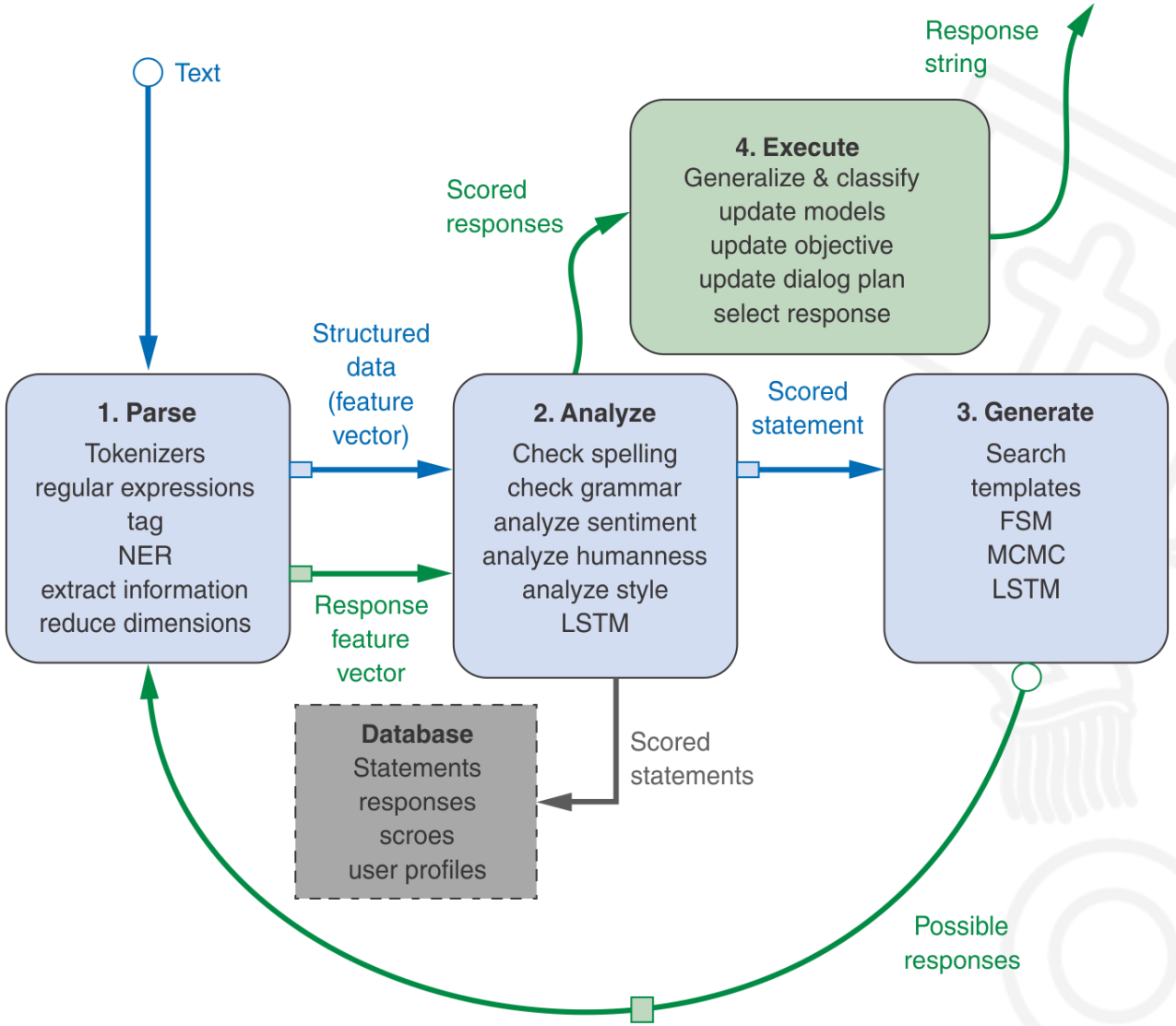
Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Linguagem natural vs. linguagem de programação

- Definição (Sistema de processamento de linguagem natural)
  - Um pipeline que envolve vários estágios de processamento, em que o input é a linguagem natural e como output temos a linguagem processada.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Sistema de NLP



Fonte: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Linguagem natural vs. linguagem de programação

- As máquinas processam linguagens desde que os computadores foram inventados.

- No entanto, essas linguagens “formais” (COBOL , Fortran, etc) foram projetadas para serem interpretadas (por um compilador ou interpretador) de uma única forma.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Linguagem natural vs. linguagem de programação

- Apesar de haver uma quantidade consideravelmente grande de linguagem de computador (mais de 700), **hoje temos um conteúdo muito maior de conteúdo em linguagem natural**.
- E o índice de documentos em linguagem natural do Google ultrapassa os **100 milhões de gigabytes** (mas estima-se que todo conteúdo online ultrapasse os **100 bilhões de gigabytes**)

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019
Ref.: https://www.google.com/search/howsearchworks/crawling-indexing/

Processamento de Linguagem Natural

- O Processamento de Linguagem Natural é um processo difícil para ser realizado por um computador.

- As máquinas ainda não podem executar tarefas como compreensão de conversação e leitura, com a mesma precisão e confiabilidade que os seres humanos.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Processamento de Linguagem Natural

- No entanto, existem técnicas poderosas o suficiente para criar máquinas que podem superar os seres humanos

- Por exemplo, hoje o computador reconhece sarcasmo de mensagens do Twitter melhor do que o ser humano. Veja você mesmo (acurácia acima de 70%):

http://www.thesarcasmdetector.com/

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019
Fonte: https://github.com/MathieuCliche/Sarcasm_detector

Aplicações práticas de Processamento de Linguagem Natural

PUC Minas Virtual

Aplicações

Tópico			
Search	Web	Documents	Autocomplete
Editing	Spelling	Grammar	Style
Dialog	Chatbot	Assistant	Scheduling
Writing	Index	Concordance	Table of contents
Email	Spam filter	Classification	Prioritization
Text mining	Summarization	Knowledge extra ction	Medical diagnoses
Law	Legal inference	Precedent search	Subpoena classification

Fonte da tabela: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Aplicações

Tópico			
News	Event detection	Fact checking	Headline composition
Attribution	Plagiarism detection	Literary forensics	Style coaching
Sentiment analysis	Community morale monitoring	Product review triage	Customer care
Behavior prediction	Finance	Election forecasting	Marketing
Creative writing	Movie scripts	Poetry	Song lyrics

Fonte da tabela: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Aplicações

- Sobre robôs jornalistas (NLP para escrita de textos):

  https://www.theverge.com/2015/1/29/7939067/ap-journalism-automation-robots-financial-reporting

- Google Duplex System

  https://ai.googleblog.com/2018/05/advances-in-semantic-textual-similarity.html

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

Processamento de Linguagem Natural

- Como um programa pode responder a uma pergunta em linguagem natural?
- Uma maneira seria usando expressões regulares:
  - expressões regulares usam um tipo especial (classe) de gramática formal, denominada gramática regular.
  - Amazon Alexa e Google Now são mecanismos baseados em padrões que utilizam gramáticas regulares.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Máquinas de Estados Finitos

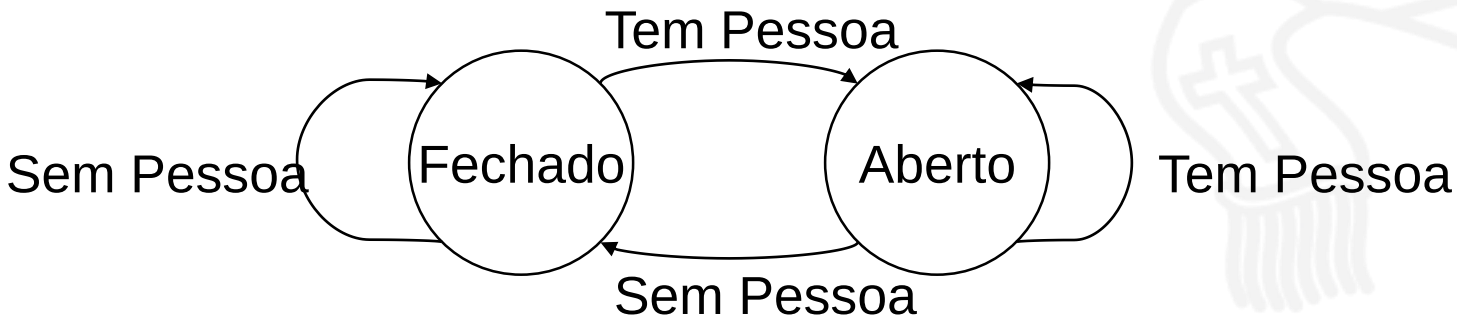
- Uma máquina que processa expressões regulares é chamada de máquina de estados finitos ou autômato finito determinístico (AFD)

- Vamos ao um exemplo (próximo slides)

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Máquinas de Estados Finitos

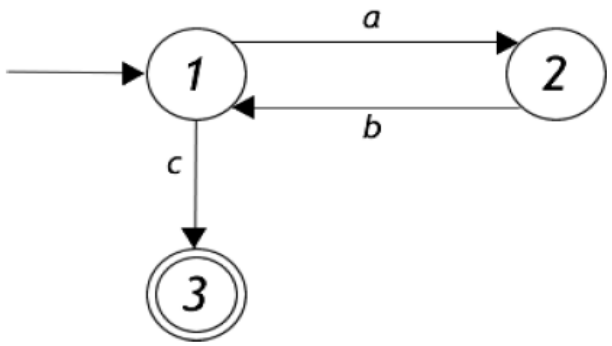
- Exemplos 1: Lâmpada
  - Dois estados: acesa ou apagada.
  - Duas ações: acender ou apagar.
- Exemplos 2: a abertura/fechamento automático de uma porta



Apenas 1 bit de memória: estado corrente do controlador

Máquinas de Estados Finitos

- Comportamento do Autômato.
  - Os apontadores indicam o estado atual da máquina (inicialmente q0). A cada transição do autômato, um símbolo da entrada e lido e os apontadores são atualizados.
  - O autômato reconhece o string de entrada se a última transição resultar em um estado final.
- AFD M para a expressão regular: (ab)*c.
- Ele aceita abc?



Máquinas de Estados Finitos

- Linguagem natural não é uma linguagem regular e não pode ser definida por uma gramatica formal

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

- Exemplo de um chatbot de correspondência de padrões (comuns antes das técnicas de machine learning) usado em sistemas como Amazon Alexa:

- Um chatbot utilizando uma máquina de estados finitos que utiliza expressões regulares para entender várias saudações.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

```
>>> import re
>>> r = "(hi|hello|hey)[ ]*([a-z]*)"
>>> re.match(r, 'Hello Rosa', flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 10), match='Hello Rosa’>

>>> re.match(r, "hi ho, hi ho, it's off to work ...", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 5), match='hi ho’>

>>> re.match(r, "hey, what's up", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 3), match='hey>
```

Biblioteca em Python para expressões regulares

https://docs.python.org/3/library/re.html

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

```
>>> import re
>>> r = "(hi|hello|hey)[ ]*([a-z]*)"
>>> re.match(r, 'Hello Rosa', flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 10), match='Hello Rosa’>

>>> re.match(r, "hi ho, hi ho, it's off to work ...", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 5), match='hi ho’>

>>> re.match(r, "hey, what's up", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 3), match='hey>
```

Expressão regular: `[a-z]*` classe de caracteres entre colchetes. O traço (-) indica um intervalo de caracteres. (*) significa uma, várias, ou nenhuma ocorrência da expressão dentro dos colchetes

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

```
>>> import re
>>> r = "(hi|hello|hey)[ ]*([a-z]*)"
>>> re.match(r, 'Hello Rosa', flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 10), match='Hello Rosa’>

>>> re.match(r, "hi ho, hi ho, it's off to work ...", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 5), match='hi ho’>

>>> re.match(r, "hey, what's up", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 3), match='hey>
```

(hi|hello|hey): “|” significa “or”. Ou seja, qualquer uma das palavras “hi”, “hello” e “hey” podem ocorrer uma única vez na frase (começando a frase por uma delas).

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

```
>>> import re
>>> r = "(hi|hello|hey)[ ]*([a-z]*)"
>>> re.match(r, 'Hello Rosa', flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 10), match='Hello Rosa’>

>>> re.match(r, "hi ho, hi ho, it's off to work ...", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 5), match='hi ho’>

>>> re.match(r, "hey, what's up", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 3), match='hey>
```

Em seguida, `[ ]*` significa um espaço em branco pode ocorrer várias vezes.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

```
>>> import re
>>> r = "(hi|hello|hey)[ ]*([a-z]*)"
>>> re.match(r, 'Hello Rosa', flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 10), match='Hello Rosa'>

>>> re.match(r, "hi ho, hi ho, it's off to work ...", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 5), match='hi ho'>

>>> re.match(r, "hey, what's up", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 3), match='hey>
```

Assim, esta expressão regular pode representar qualquer sentença que comece com "hi", "hello" ou "hey", seguida de zero ou mais white spaces e uma palavra com qualquer combinação de "a" a "z" minúsculos.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

```
>>> import re
>>> r = "(hi|hello|hey)[ ]*([a-z]*)"
>>> re.match(r, 'Hello Rosa', flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 10), match='Hello Rosa’>

>>> re.match(r, "hi ho, hi ho, it's off to work ...", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 5), match='hi ho'>

>>> re.match(r, "hey, what's up", flags=re.IGNORECASE)
<_sre.SRE_Match object; span=(0, 3), match='hey>
```


Ignora caracteres maiúsculos e minúsculos (para simplificar)

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

- Vamos tornar a expressão regular poderosa para tentar reconhecer mais saudações (próximo slide):

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

```
>>> r = r"[^a-z]*([y]o|[h']?ello|ok|hey|(good[ ])?(morn[gin']{0,3}|"\
...     r"afternoon|even[gin']{0,3}))[\s,;:]{1,3}([a-z]{1,20})"
>>> re_greeting = re.compile(r, flags=re.IGNORECASE)
>>> re_greeting.match('Hello Rosa’)
<_sre.SRE_Match object; span=(0, 10), match='Hello Rosa’>

>>> re_greeting.match('Hello Rosa').groups()
('Hello', None, None, 'Rosa’)

>>> re_greeting.match("Good morning Rosa")
<_sre.SRE_Match object; span=(0, 17), match="Good morning Rosa">
```

O `r` antes da sentença indica um texto bruto (não uma expressão regular)

O compilador da expressão regular irá traduzir “r” para uma expressão regular.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

```
>>> re_greeting.match("Good Manning Rosa")
>>> re_greeting.match('Good evening Rosa Parks').groups()
('Good evening', 'Good ', 'evening', 'Rosa’)

>>> re_greeting.match("Good Morn'n Rosa")
<_sre.SRE_Match object; span=(0, 16), match="Good Morn'n Rosa">

>>> re_greeting.match("yo Rosa")
<_sre.SRE_Match object; span=(0, 7), match='yo Rosa'>
```

Não reconhece erros de digitação

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Um chatbot com expressão regular

Expressões Regulares

- Devido a tantas limitações (e um trabalho muito tedioso), compor expressões regulares à mão não é a única maneira de treinar um chatbot.

- Iremos concluir o chatbot

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

```
import re

r = r"[^a-z]*([y]o|[h']?ello|ok|hey|(good[ ])?(morn[gin']{0,3}|"\
    r"afternoon|even[gin']{0,3}))[\s,;:]{1,3}([a-z]{1,20})"

re_greeting = re.compile(r, flags=re.IGNORECASE)

my_names = set(['rosa', 'rose', 'chatty', 'chatbot', 'bot', 'chatterbot'])
curt_names = set(['hal', 'you', 'u'])
greeter_name = ''
match = re_greeting.match(input())

if match:
    at_name = match.groups()[-1]
    if at_name in curt_names:
        print("Good one.")
    elif at_name.lower() in my_names:
        greeter_name = at_name
        print("Hi {}, How are you?".format(greeter_name))
```

Fonte da tabela: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

- Execute o script do chatbot e tente entrar as seguintes sentenças:

  hello rosa hey you

  e veja como ele se comporta

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

- Os primeiros trabalhos em NLP projetaram manualmente regras lógicas para extrair informações de uma sequência de linguagem natural (abordagem baseada em padrões - pattern-based approach)

- Estes padrões não precisam ser padrões de sequência de caracteres, como nosso exemplo de expressão regular. NLP também pode envolver padrões de sequências de palavras.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Expressões Regulares

- Os principais componentes da PNL, como stemmers e tokenizers, bem como sofisticados mecanismos de diálogo de ponta a ponta (ELIZA[1]), foram criados dessa maneira, a partir de expressões regulares e correspondência de padrões.

[1] Primeiro chatbot desenvolvido (1966): https://www.masswerk.at/elizabot/
Weizenbaum, Joseph "ELIZA – A Computer Program For the Study of Natural Language Communication Between Man and Machine"
in: Communications of the ACM; Volume 9 , Issue 1 (January 1966): p 36-45

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Pipeline Processamento de Linguagem Natural

PUC Minas Virtual

Processamento de Linguagem Natural

- Alternativa ao uso da abordagem baseada em padrões: a construção de um banco de dados contendo sessões de diálogo entre humanos, declarações e respostas para milhões de conversas. Seria viável?

- Neste caso, para um chatbot, seria procurar nesse banco a mesma sequência exata de caracteres que o usuário "disse" ao chatbot.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Processamento de Linguagem Natural

- Problema desta abordagem: um único erro de digitação ou variação na declaração atrapalharia o chatbot.

- Para solução deste tipo de problema, podemos usar uma abordagem baseada distância (correspondência) de caracteres:
  - Problema: medir distância entre sentenças em linguagem natural tende a obter resultados errados: sentenças similares com “bom” e “okay”, podem ter diferentes caracteres (distância medida será alta).

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Processamento de Linguagem Natural

- No entanto, estas métricas baseadas em vetores são úteis em algumas aplicações de NLP, como corretores ortográficos e reconhecimento de nomes próprios (melhor para ortografias e não tanto para significado).

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Processamento de Linguagem Natural

- Outra possibilidade é o uso de métricas de distância entre vetores (Jaccard distance, Euclidean distance, etc.).
  - Problema: pode adicionar distorções e atrapalhar o chatbot quando se depara com erros ortográficos.

- Essas métricas falham em capturar a essência do relacionamento entre duas cadeias de caracteres quando são diferentes (elas tendem a dar bom resultado quando a cadeia de caractere é muito semelhante).

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Pipeline para chatbot de linguagem natural

- O pipeline de PNL necessário para criar um chatbot é semelhante ao pipeline necessário para criar um sistema de resposta a perguntas.
- Um chatbot requer quatro tipos de processamento e um banco de dados para manter a memória de declarações e respostas anteriores.

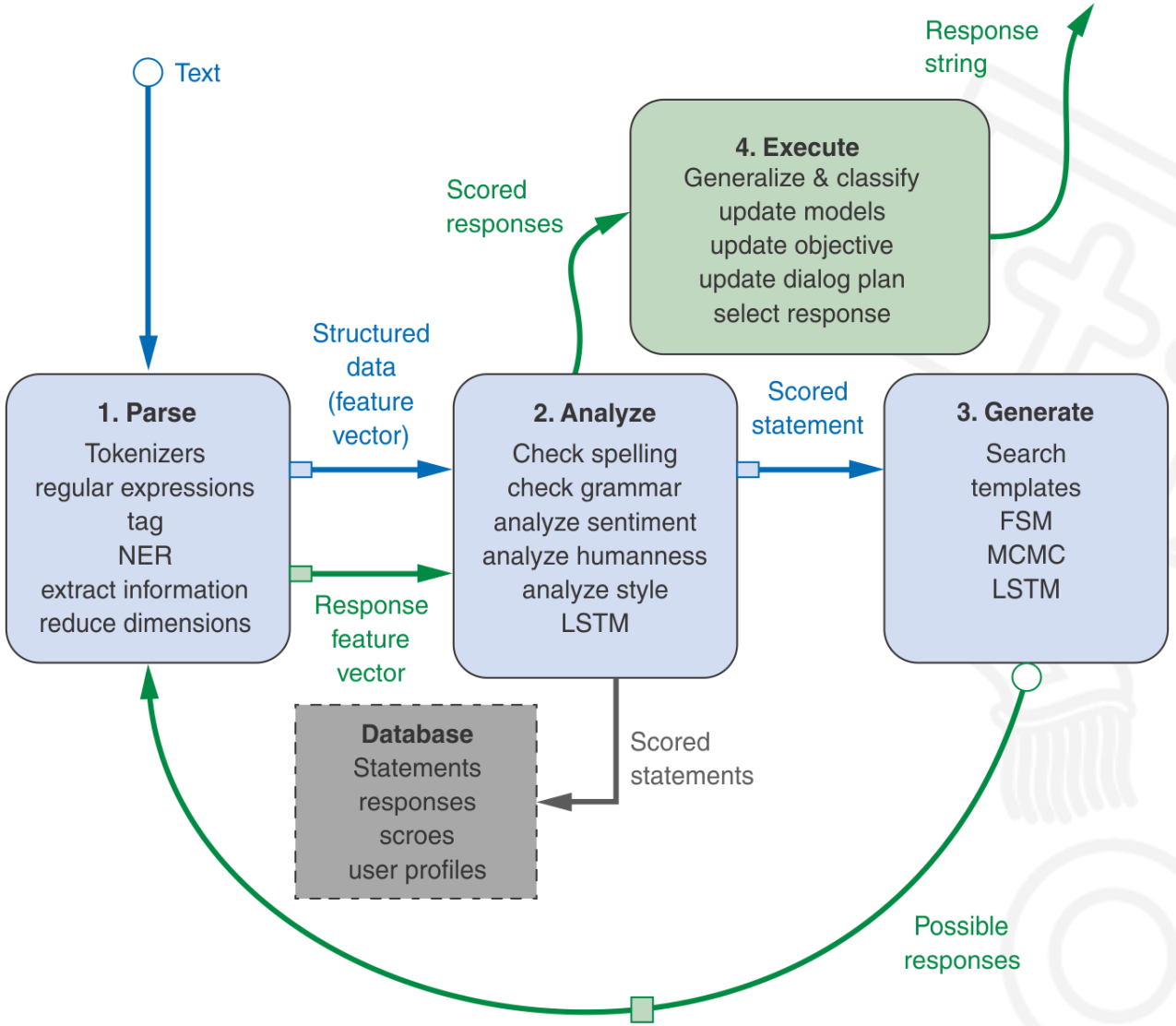
Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Pipeline para chatbot de linguagem natural

- Cada um dos quatro estágios de processamento pode conter um ou mais algoritmos de processamento trabalhando em paralelo ou em série (figura no próximo slide)

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Sistema de NLP



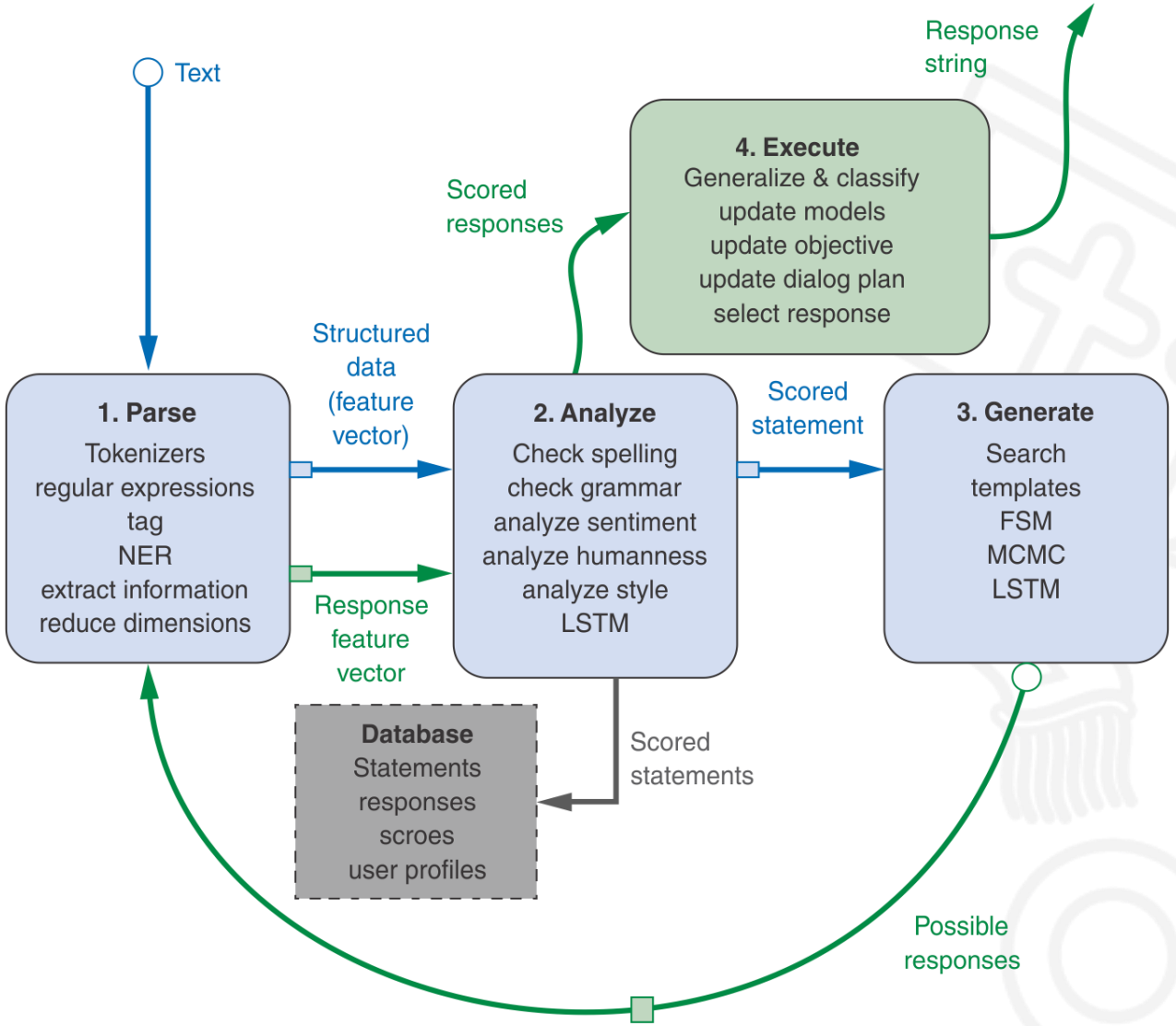
Fonte: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Sistema de NLP

Cada um desses estágios pode ser implementado usando um ou mais dos algoritmos listados em cada fase

1. Parse: Extrai recursos, dados numéricos estruturados, de texto em idioma natural.

2. Analyze: Gera e combina features classificando o texto quanto a sentimentos



4. Execute: planeja declarações com base no histórico e nos objetivos da conversa e seleciona a próxima resposta.

3. Generate: produz possíveis respostas usando modelos, busca ou modelos de idioma.

Fonte: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Considerações Finais

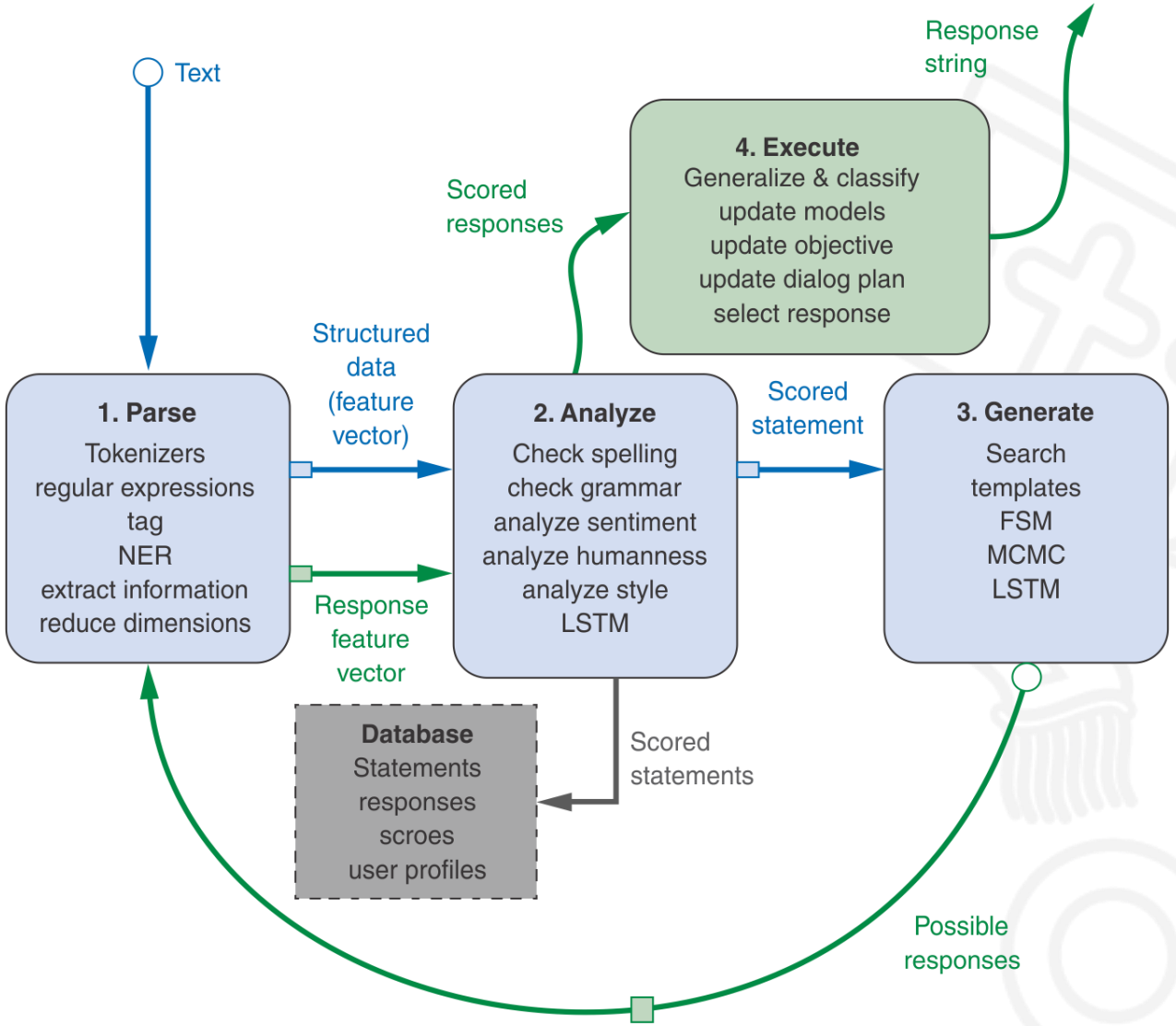
PUC Minas Virtual

Sistema de NLP

Cada um desses estágios pode ser implementado usando um ou mais dos algoritmos listados em cada fase

1. Parse: Extrai recursos, dados numéricos estruturados, de texto em idioma natural.

2. Analyze: Gera e combina features classificando o texto quanto a sentimentos



4. Execute: planeja declarações com base no histórico e nos objetivos da conversa e seleciona a próxima resposta.

3. Generate: produz possíveis respostas usando modelos, busca ou modelos de idioma.

Fonte: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Próximos tópicos

- Algoritmos e técnicas de processamento em linguagem natural.
- Expressões regulares. Medidas de similaridade textual (TF-IDF). Parsing, tokenização, lematização, stemming.
- Extração de informação. Arquitetura de aplicação para processamento de Linguagem Natural.
- Análise de sentimento.

Ref.: Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action** 2019

Dicas!

Links para saber mais:

Detector de Sarcasmo

http://www.thesarcasmdetector.com/

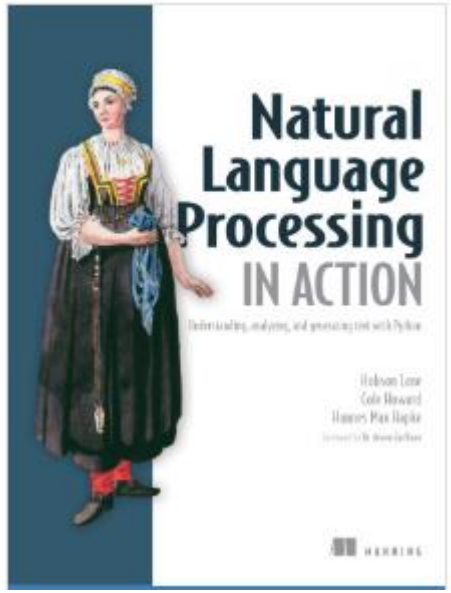
Dicas!

Links para saber mais:

Google Duplex System

https://ai.googleblog.com/2018/05/advances-in-semantic-textual-similarity.html

Principais Referências



Hobson Lane, Cole Howard, Hannes Hapke. **Natural Language Processing in Action:Understanding, analyzing, and generating text with Python**. March 2019

PUC Minas
Virtual