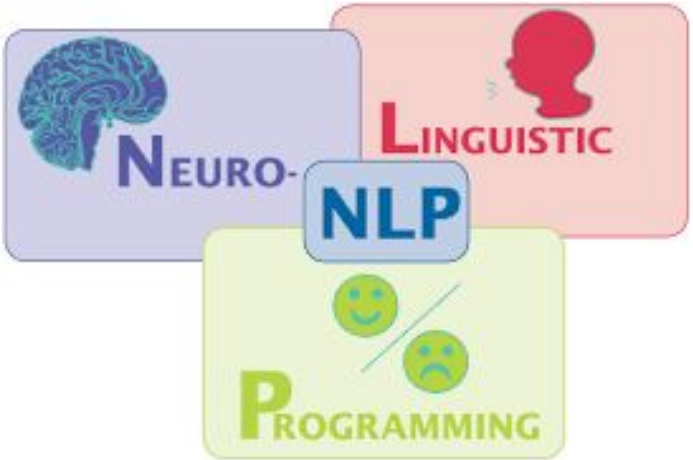
Dados não estruturados

PLN - Aula 12

Objetivos

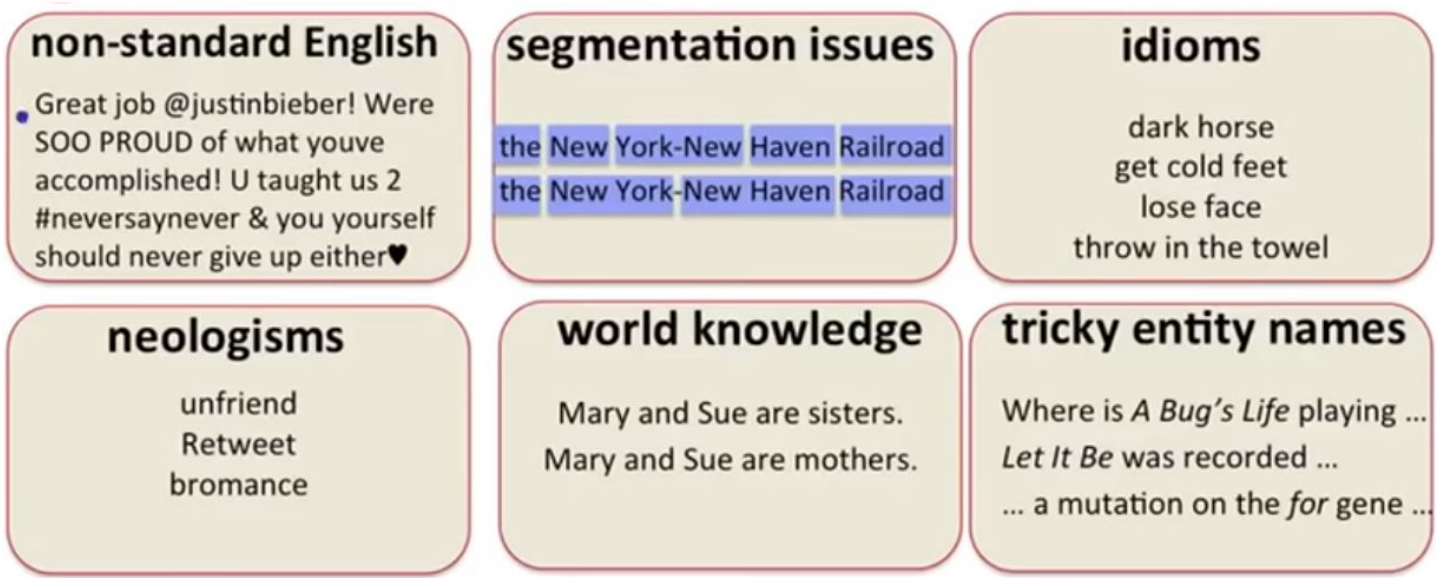
- Introdução
- Aplicações
- Pré processamento
- PLN em Python: SpaCy e NLTK
- Partes do discurso (POS - Parts Of Speech)

O que é PNL?

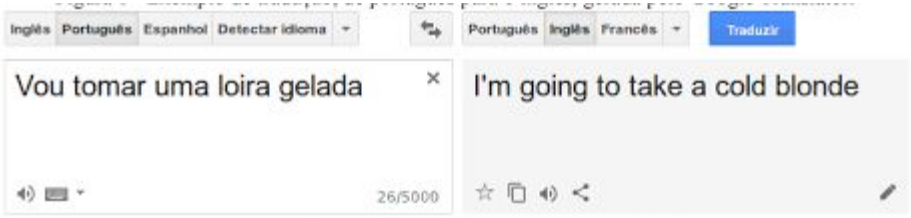


Objetivos: “Entender” a linguagem natural humana

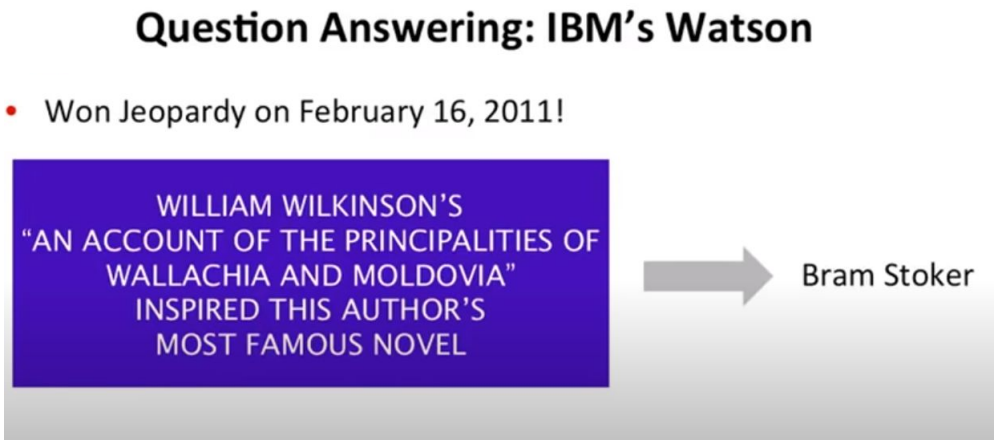
Desafios


non-standard English
Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either♥
segmentation issues
the New York-New Haven Railroad
the New York-New Haven Railroad
idioms
dark horse
get cold feet
lose face
throw in the towel
neologisms
unfriend
Retweet
bromance
world knowledge
Mary and Sue are sisters.
Mary and Sue are mothers.
tricky entity names
Where is A Bug's Life playing ...
Let It Be was recorded ...
... a mutation on the for gene ...

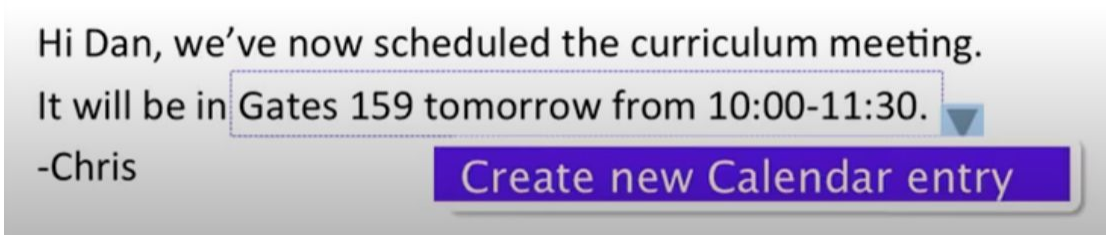
Aplicações: Tradução Automática



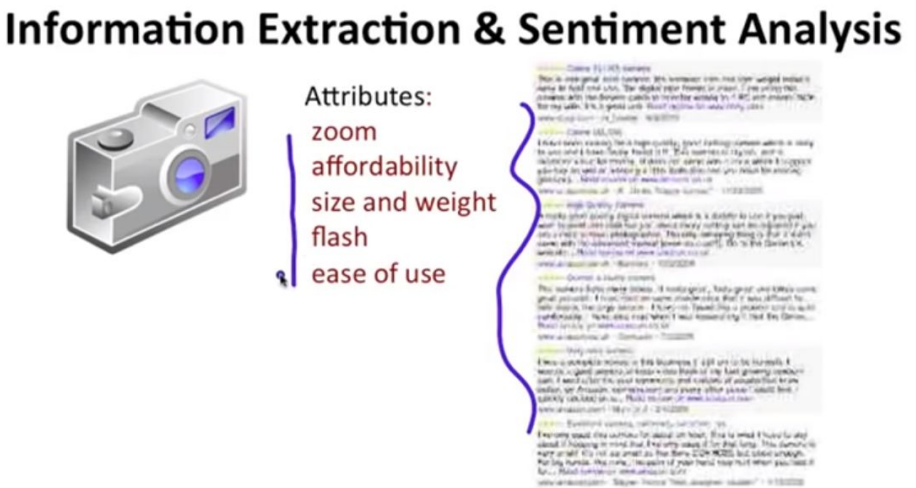
Pergunta e resposta



Aplicações: Extração contextual

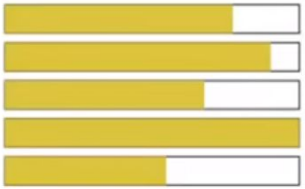


Aplicações: Análise de sentimentos e extração informação



Aplicações: Análise de sentimentos e extração informação

Attributes:



Size and weight

- ✓ nice and compact to carry!
- ✓ since the camera is small and light, I won't need to carry around those heavy, bulky professional cameras either!
- ✗ the camera feels flimsy, is plastic and very light in weight you have to be very delicate in the handling of this camera

Aplicações: Detecção de Spam

Estado da arte: bem resolvidos

Estado da arte: em progresso

Estado da arte: Difíceis

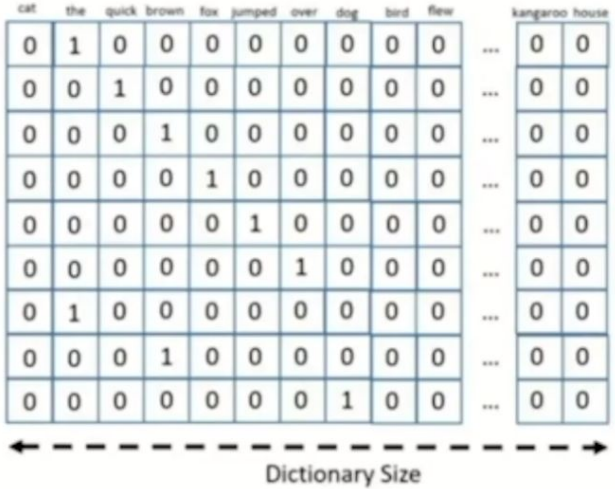
Estado da arte

https://nlpprogress.com/

Processo clássico: Aprendizado supervisionado

Extração de feature: Representação vetorial (bag of words)

Extração de feature: Representação vetorial (bag of words)



SpaCy e NLTK

Instalando

pip install -U spacy

python -m spacy download pt_core_news_sm # para o modelo em português

python -m spacy download en-core-web-sm # para o modelo em inglês

Pip install -U nltk

Atividades de NLP: Baixo nível

Tokenização — Divisão de uma string em listas de pedaços ou “tokens”.

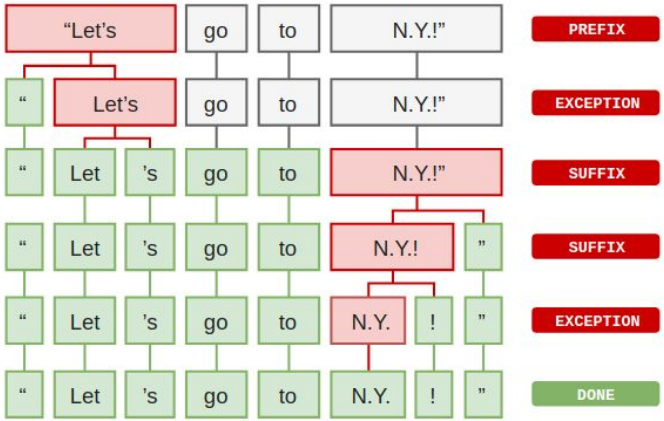
Remoção de Stopwords — Remoção de palavras comuns que normalmente não contribuem para o significado de uma frase

Stemming — Remoção de sufixos e prefixos de uma palavra

Lemmatization — Determinação do lema para uma determinada palavra.

Part-of-Speech Tagging — Etiquetagem de elementos textuais, com o fim de evidenciar a estrutura gramatical de um determinado trecho de texto.

Pré processamento: Tokenização

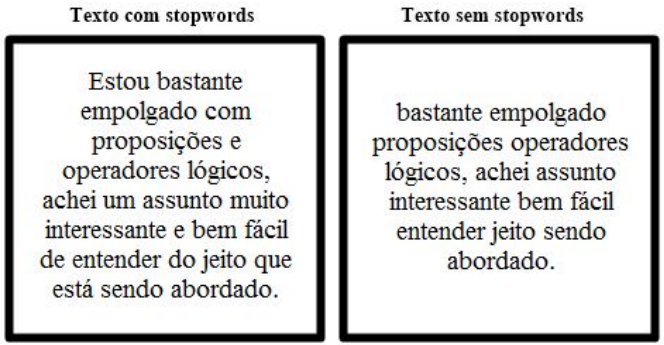


Exercício

1) Importar o spacy
2) Carregar o modelo pt_core_news_sm com: nlp = spacy.load("pt_core_news_sm") (https://spacy.io/usage/models)
3) Vamos criar um objeto Doc:
    a) doc = nlp(u'Esse texto é meramente ilustrativo.')

Pré processamento: Remoção de Stop Words

Palavras que não tem relevância para aplicação que deseja trabalhar, ex: problema na frequência de palavras, ex: as , e, para, com, etc



Exercício

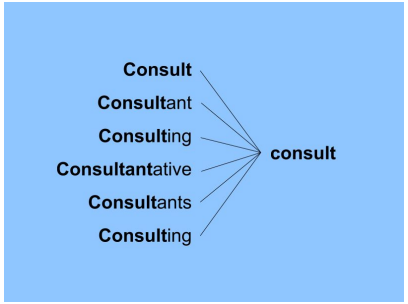
1) Carregar a lista de stop words: spacy_stopwords = spacy.lang.pt.stop_words.STOP_WORDS

2) Contar quantas stop words existem no modelo

3) Mostrar algumas dessas stop words

Pré processamento: Stemização e Lematização

Steamização (stemming):
- Reduzir a palavra à sua raiz



Lematização (Lemmatization):
- Reduzir a palavra à sua base

Exercício: Stemming

1) Importar o nltk
2) Importar o stem snowball: from nltk.stem.snowball import SnowballStemmer
3) Criar a classe stemmer = SnowballStemmer(language='portuguese')
4) Definir: doc = nlp(u'computar computado computando computador')
5) Aplicar o stemmer.steam(token)

Exercício: Lemmatization

1) Com nosso doc, aplicar extrair o atributo lemma_ em cada token

Part-of-Speech Tagging

O spacy oferece uma ferramenta de classificação gramatical, que permite classificar determinado token, dentro do contexto do texto, entre as possíveis classes gramaticais, como por exemplo:

- Substantivos (noun)
- Verbos (verb)
- Adjetivos (adjective)
- Pontuação (punctuation);
- etc.

Para isso, usamos os seguintes métodos:

- .orth_: retorna o token em string;
- .pos_: retorna a classe gramatical do token;
- .tag_: retorna uma tag com explicações contextuais acerca da classificação;
- spacy.explain(token.tag_) se disponível, é a função que formula uma explicação da tag, que, como podemos ver, não é muito evidente. Se não disponível, a função retorna "None"

Exercício

1) Na nossa frase, vamos extrair as partes do discurso com o atributo de cada token.pos_
2) Com o texto fornecido, Filtrar stop words
3) Lematizar as palavras
4) verificar a frequência de cada POS Tag , Ex:
    a) VERB: 5
    b) NOUN: 6