UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Minicurso: Topic Modeling using Transformer

Processamento de Linguagem Natural

Professor: Michael Oliveira da Cruz

UFRPE, 07 Dezembro 2023

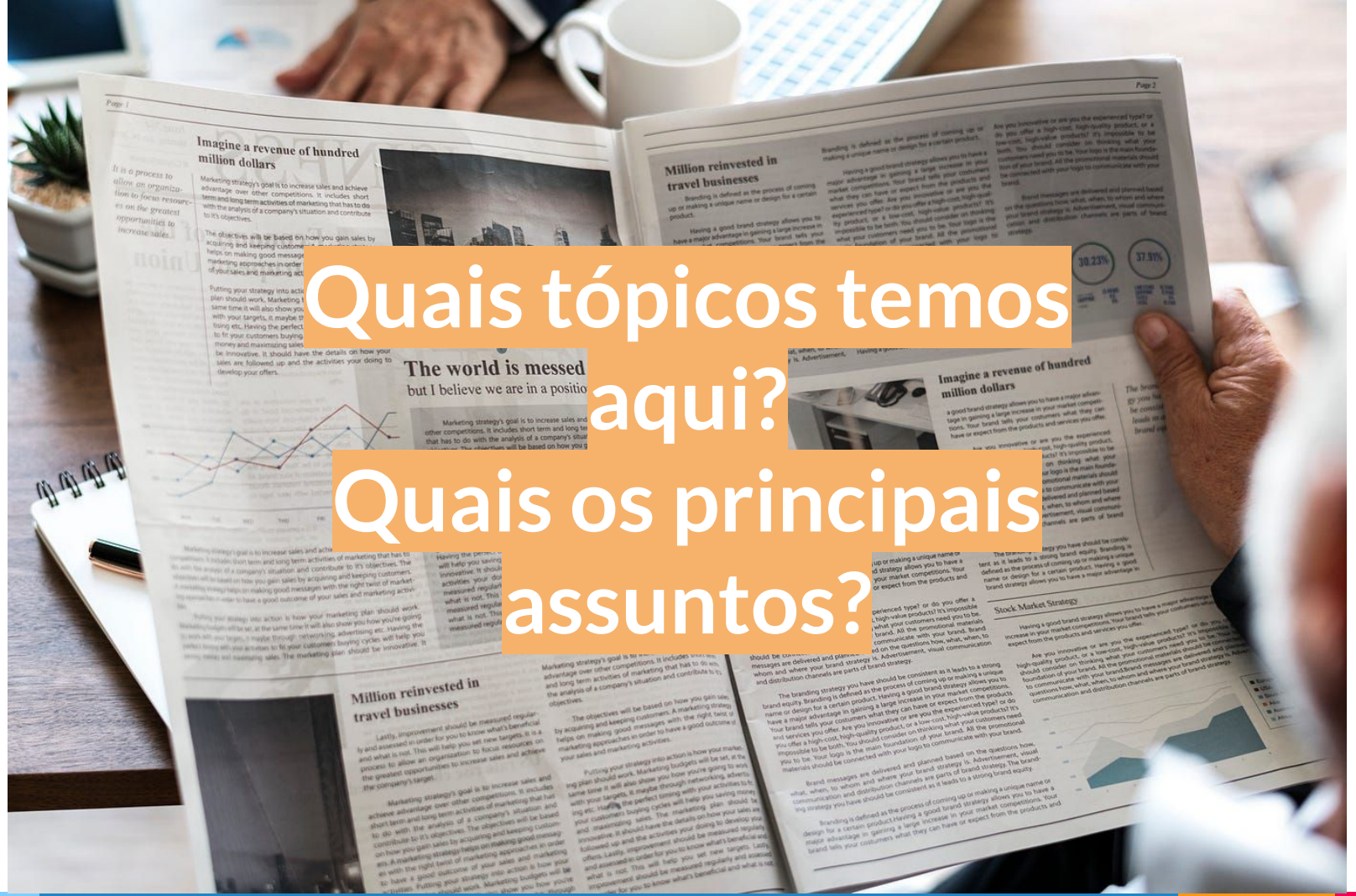
Agenda

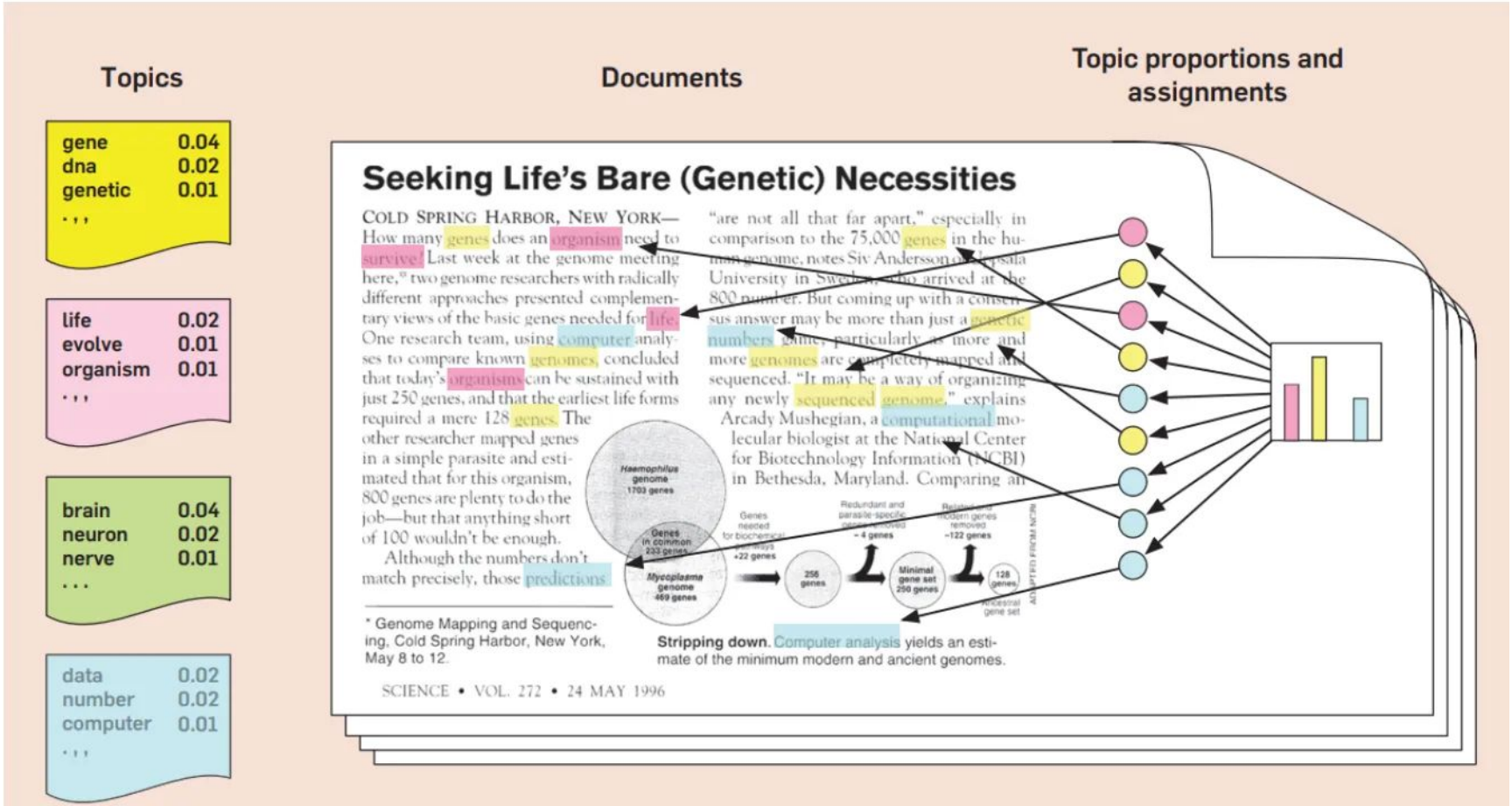
1. Motivação
2. Introdução a pré-processamento em NLP
3. Representação
4. LDA (Latent Dirichlet Allocation)
5. BERTopic
6. Hugging Face

1.
Motivação

Por que precisamos encontrar tópicos?


Imagine a revenue of hundred million dollars
The world is messed up
but I believe we are in a position now to help
Million reinvested in travel businesses
Fact about online marketing
Stock Market Strategy


Quais tópicos temos aqui?
Quais os principais assuntos?


Topics
gene 0.04
dna 0.02
genetic 0.01
. . .
life 0.02
evolve 0.01
organism 0.01
. . .
brain 0.04
neuron 0.02
nerve 0.01
. . .
data 0.02
number 0.02
computer 0.01
. . .
Documents
Topic proportions and assignments
Seeking Life's Bare (Genetic) Necessities
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.
Although the numbers don't match precisely, those predictions
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
Redundant and parasite-specific genes ~4 genes
Relatively modern genes removed ~122 genes
256 genes
Minimal gene set 250 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.
SCIENCE • VOL. 272 • 24 MAY 1996

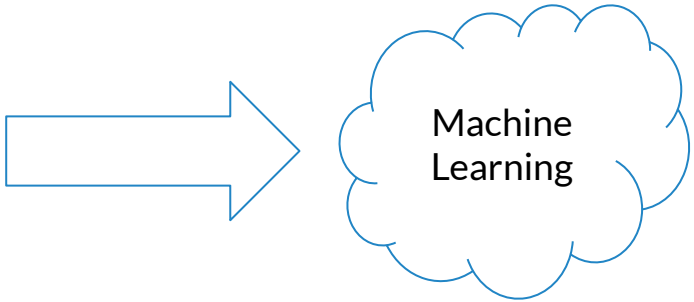
2.

Introdução a Pré-Processamento

Conceitos Básicos

Restaurant Reviews

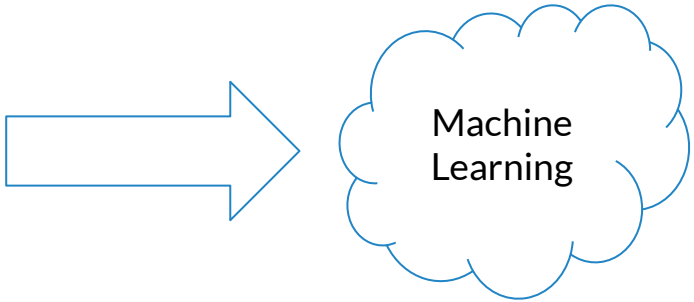
Now I am getting angry and I want my damn pho.
Honeslty it didn't taste THAT fresh.)
The potatoes were like rubber and you could tell they had been made up ahead of time being kept unde...
The fries were great too.
A great touch.
Service was very prompt.
Would not go back.
The cashier had no care what so ever on what I had to say it still ended up being wayyy overpriced.
I tried the Cape Cod ravoli, chicken, with cranberry...mmmm!



Restaurant Reviews

Now I am getting angry and I want my damn pho.
Honeslty it didn't taste THAT fresh.)
The potatoes were like rubber and you could tell they had been made up ahead of time being kept unde...
The fries were great too.
A great touch.
Service was very prompt.
Would not go back.
The cashier had no care what so ever on what I had to say it still ended up being wayyy overpriced.
I tried the Cape Cod ravoli, chicken, with cranberry...mmmm!

How to pass text to a machine learning model?



Problemas

- Letras maiúsculas e minúsculas

Problemas

- Letras maiúsculas e minúsculas
- Sinais de pontuação

Problemas

- Letras maiúsculas e minúsculas
- Sinais de pontuação
- Palavras com números ou números

Problemas

- Letras maiúsculas e minúsculas
- Sinais de pontuação
- Palavras com números ou números
- Espaços duplicados

Is that all?

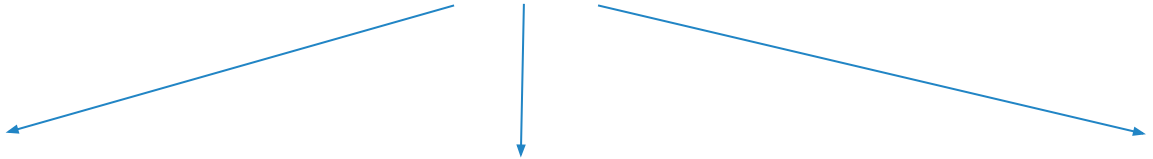
Tokenization

- É uma das primeiras etapas de pré-processamento
  - A tokenização quebra o texto em pedaços de informação que será considerada como discreta.

  - A tokenização pode ser no nível:
    - Palavras
    - Sentenças
    - Caracteres/letras
    - Subpalavras

Tokenization

"Sucesso é o acúmulo de pequenos esforços, repetidos dia e noite"



['Sucesso', 'é', 'o', 'acúmulo', 'de', 'pequenos', 'esforços', 'repetidos', 'dia', 'e', 'noite']

Stop-words

- São palavras do que possuem um alto grau de repetição no texto e que não acrescentam informação:
  - Artigos
  - Preposição
  - Pronomes
  - Etc etc etc

Stop-words

a, agora, ainda, alguém, algum, alguma, algumas, alguns, ampla, amplas, amplo, amplos, ante, antes, ao, aos, após, aquela, aquelas, aquele, aqueles, aquilo, as, até, através, cada, coisa, coisas, com, como, contra, contudo, da, daquele, daqueles, das, de, dela, delas, dele, deles, depois, dessa, dessas, desse, desses, desta, destas, deste, deste, destes, deve, devem, devendo, dever, deverá, deverão, deveria, deveriam, devia, deviam, disse, disso, disto, dito, diz, dizem, do, dos, e, é, ela, elas, ele, eles, em, enquanto, entre, era, essa, essas, esse, esses, esta, está, estamos, estão, estas, estava, estavam, estávamos, este, estes, estou, eu, fazendo, fazer, feita, feitas, feito, feitos, foi, for, foram, fosse, fossem, grande, grandes, há, isso, isto, já, la, lá, lhe, lhes, lo, mas, me, mesma, mesmas, mesmo, mesmos, meu, meus, minha, minhas, muita, muitas, muito, muitos, na, não, nas, nem, nenhum, nessa, nessas, nesta, nestas, ninguém, no, nos, nós, nossa, nossas, nosso, nossos, num, numa, nunca, o, os, ou, outra, outras, outro, outros, para, pela, pelas, pelo, pelos, pequena, pequenas, pequeno, pequenos, per, perante, pode, pude, podendo, poder, poderia, poderiam, podia, podiam, pois, por, porém, porque, posso, pouca, poucas, pouco, poucos, primeiro, primeiros, própria, próprias, próprio, próprios, quais, qual, quando, quanto, quantos, que, quem, são, se, seja, sejam, sem, sempre, sendo, será, serão, seu, seus, si, sido, só, sob, sobre, sua, suas, talvez, também, tampouco, te, tem, tendo, tenha, ter, teu, teus, ti, tido, tinha, tinham, toda, todas, todavia, todo, todos, tu, tua, tuas, tudo, última, últimas, último, últimos, um, uma, umas, uns, vendo, ver, vez, vindo, vir, vos, vós

Lemmatization

- É o processo para reduzir uma palavra para sua raiz ou para uma forma padrão.
  - Essa técnica é usada para diminuir as variações que uma palavra pode ter
  - Exemplo: tivemos, tive -> ter
  - Exemplo: amigo, amiga, amigão -> amigo

Lemmatization

Fonte: https://www.turing.com/kb/stemming-vs-lemmatization-in-python

3.

Data Representation

One Hot Encoding, BoW, Embeddings

One hot encoding

- É uma técnica que transforma cada palavra em um vetor contendo 0 ou 1.
- Essa técnica tem a característica de ser esparsa, visto que o tamanho do vetor é o tamanho do vocabulário
- Uma sentença é composta por um conjunto de vetores

One hot encoding

The cat sat on the mat

The: [0 1 0 0 0 0 0]

cat: [0 0 1 0 0 0 0]

sat: [0 0 0 1 0 0 0]

on: [0 0 0 0 1 0 0]

the: [0 0 0 0 0 1 0]

mat: [0 0 0 0 0 0 1]

Bag Of Words (BoW)

- É uma técnica para converter dados textuais em dados numéricos

- A ideia é criar uma "bag" de palavras
  - Esse saco de palavras servirá para definir o tamanho do vetor de sentenças
  - Cada valor do vetor representa a frequência de uma palavra
- Aqui também temos a presença de vetores esparsos

Bag Of Words (BoW)

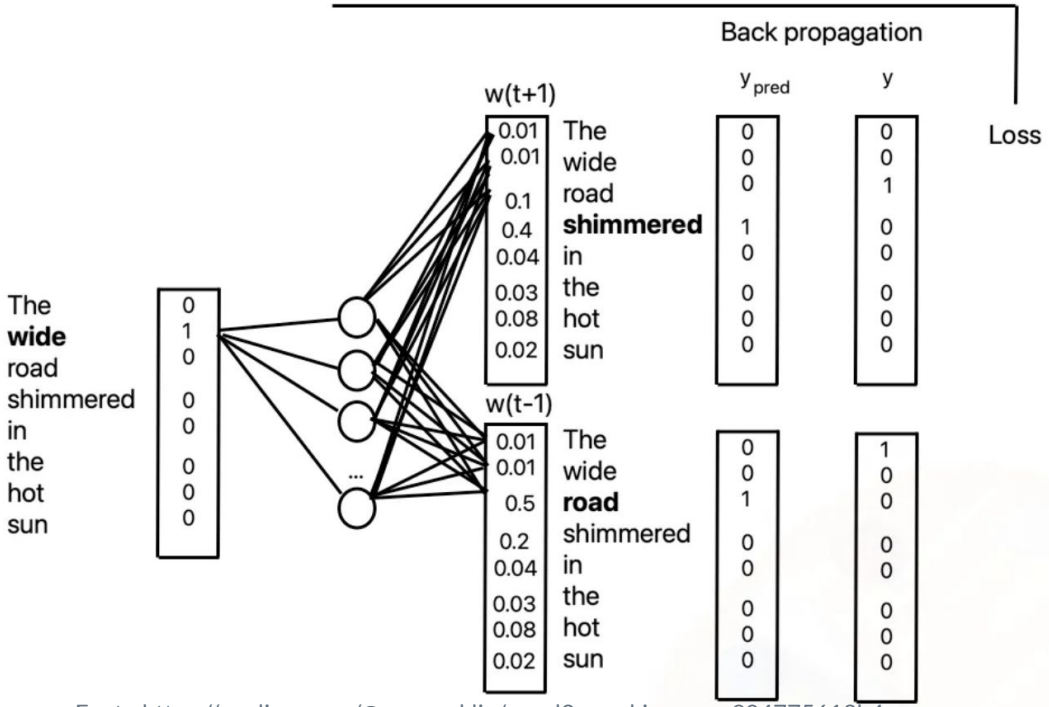
```
corpus = [
...     'This is the first document.',
...     'This document is the second
document.',
...     'And this is the third one.',
...     'Is this the first document?',
... ]
```

```
array(['and', 'document', 'first', 'is', 'one', 'second', 'the', 'third',
       'this'], ...)
>>> print(X.toarray())
[[0 1 1 1 0 0 1 0 1]
 [0 2 0 1 0 1 1 0 1]
 [1 0 0 1 1 0 1 1 1]
 [0 1 1 1 0 0 1 0 1]]
```

Embeddings

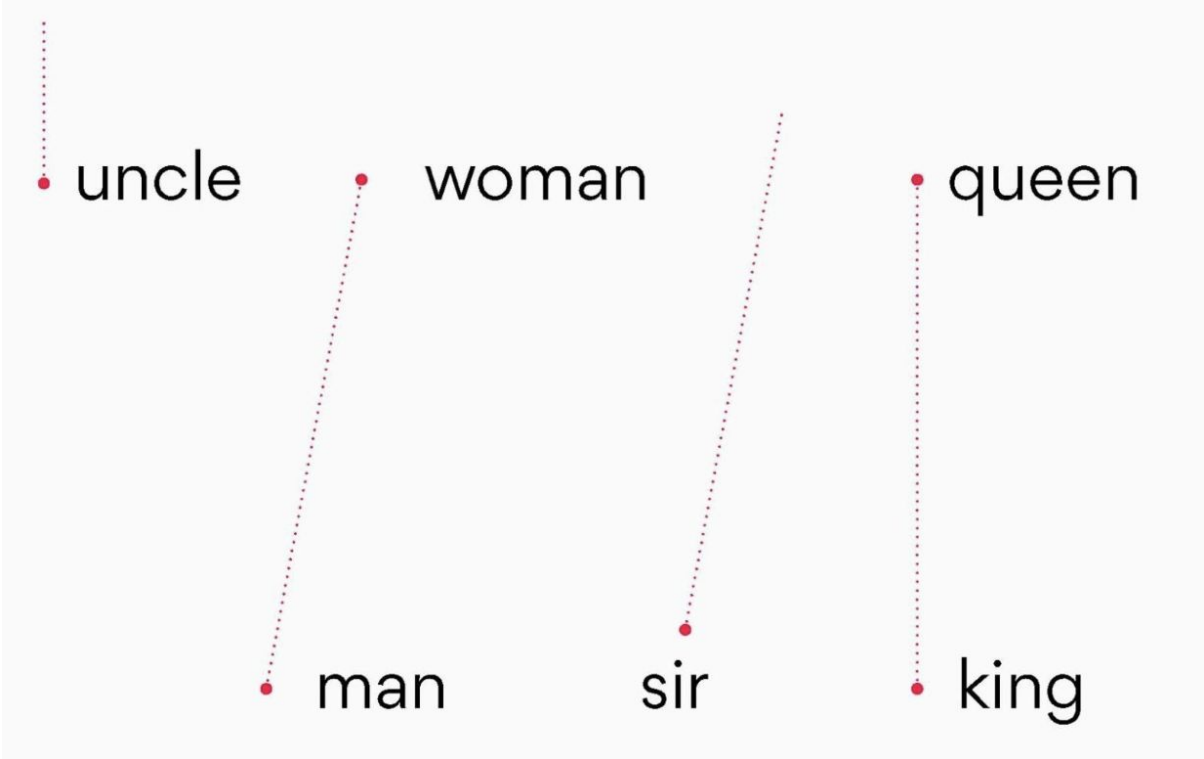
- É uma técnica que gera vetores para cada palavra
  - Esses vetores são de tamanhos pré-definidos
  - São vetores densos
  - Esses vetores contêm informações de semânticas/proximidades entre palavras

Embeddings



Fonte:https://medium.com/@corymaklin/word2vec-skip-gram-904775613b4c

Embeddings



Fonte: https://www.turing.com/kb/guide-on-word-embeddings-in-nlp

4.

LDA - Latent Dirichlet Allocation

Uma Visão Geral

LDA

- Tópicos ou temas são basicamente um conjunto de palavras que são estatisticamente relevante dentro de um *corpus*

- A ideia básica da modelagem de tópicos é considerar:
  - Um texto é formado por uma distribuição de tópicos
  - Tópico é formado por uma distribuição de palavras

LDA

- A ideia do LDA para encontrar tópicos é a seguinte:
  - Dado um corpus formado por M documentos:
    - D1: As pessoas gostam muito de esportes
    - D2: As mudanças climáticas podem ameaçar o futuro das nossas crianças.
    - D3: O Liverpool está muito próximo de conseguir mais uma taça
  - Cria-se uma Matriz de Palavras dos Documentos

LDA

Document Word Matrix

	W1	W2	W3	W4	W5	W6	W7	W8
D1	0	1	1	0	1	1	0	1
D2	1	1	1	1	0	1	1	0
D3	1	0	0	0	1	0	0	1
D4	1	1	0	1	0	0	1	0
D5	0	1	0	1	0	0	1	0

LDA

- A ideia do LDA para encontrar tópicos é a seguinte:
  - A partir da MPD, cria-se duas outras matrizes:
    - Matriz de Tópicos Por Documentos
    - Matriz de Palavras Por Tópicos

Document Topic Matrix

	K1	K2	K3	K4	K5	K6
D1	1	0	0	0	0	0
D2	0	1	0	0	1	1
D3	1	1	0	0	0	0
D4	1	0	0	1	0	1
D5	0	0	1	1	0	0

Shape: 5 * 6

Topic Word Matrix

	W1	W2	W3	W4	W5	W6	W7	W8
K1	0	1	1	0	1	0	1	0
K2	1	1	1	1	0	1	1	1
K3	1	0	0	0	0	1	0	0
K4	1	1	0	1	1	0	0	1
K5	0	0	1	1	0	1	1	1
K6	1	0	1	1	1	0	0	1

Shape: 6 * 8

- O objetivo da técnica LDA é gerar a representação ótima dessas duas matrizes.

LDA

- Quais tópicos geram um documento?
- Quais palavras geram um tópico?

LDA

- LDA usa um processo iterativo:
  - Primeiramente atribui aleatoriamente tópicos às palavras dos documentos.
  - Logo, cada palavra poderá estar relacionada a mais de um tópico

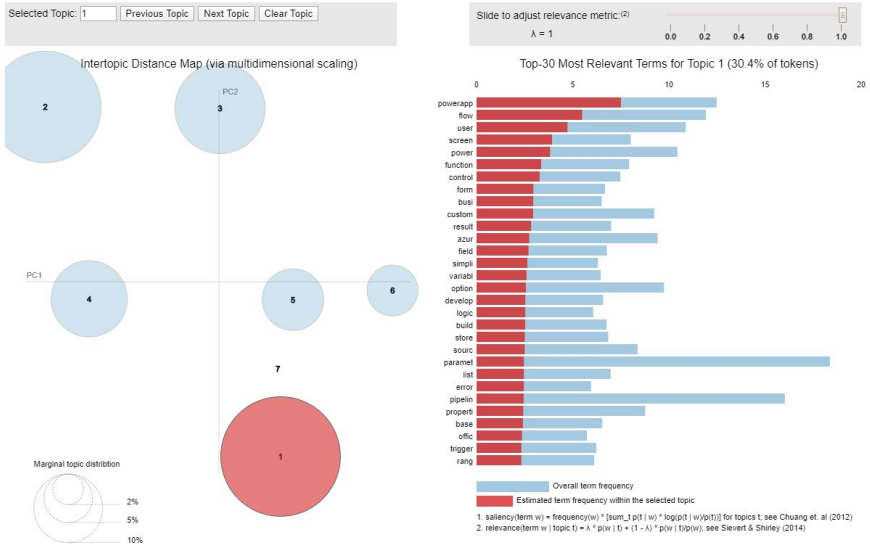
- D1 = (w1 (k5), w2 (k3), w3 (k1), w4 (k2), w5 (k5), w6 (k4), w7 (k7), w8(k1))
- D2 = (w`1(k2), w`2 (k4), w`3 (k2), w`4 (k1), w`5 (k2), w`6 (k1), w`7 (k5), w`8(k3), w`9 (k7), w`10(k1))
- D3 = (w“1(k3), w“2 (k1), w“3 (k5), w“4 (k3), w“5 (k4), w“6(k1),..., w“13 (k1), w“14(k3), w“15 (k2))
- D4 = (w“`1(k4), w“`2 (k5), w“`3 (k3), w“`4 (k6), w“`5 (k5), w“`6 (k3) ..., w“`10 (k3), w“`11 (k7), w“`12 (k1))
- D5 = (w““1 (k1), w““2 (k7), w““3 (k2), w““4 (k8), w““5 (k1), w““6(k8) ..., w““32(k3), w““33(k6), w““34 (k5))

LDA

- LDA usa um processo iterativo:
  - A partir da distribuição anterior podemos gerar as duas matrizes de forma inicial
  - Assim, para cada Documento para cada palavra (w):
    - Iteramos sobre cada tópico para calcular:
      - p_1 = a proporção de palavras no documento D que estão atribuídas ao tópico k_i
      - p_2 = a proporção de documentos em que w_j é atribuído para k_i
      - Usando o produto de $p_1 x p_2$ atribuímos um novo tópico para w_j

LDA

- LDA usa um processo iterativo:
  - A iteração continua até que se alcance uma estabilidade nas mudanças de tópicos



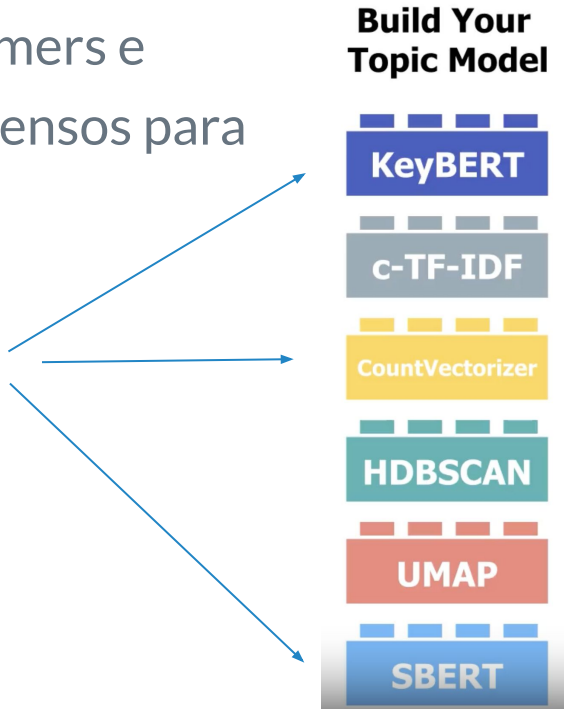
4.
BERTopic

Visão Geral

BERTopic

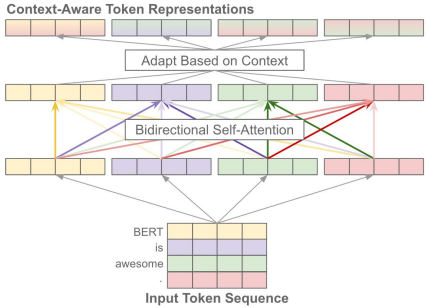
- É uma técnica baseada em Transformers e c-TF-IDF para criar agrupamentos densos para facilitar a interpretação de tópicos.

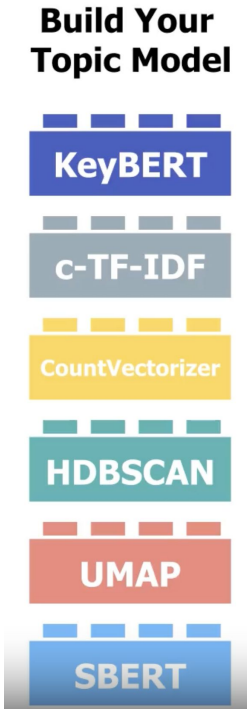
- A técnica é baseada nesses pilares



BERTopic

- Bert é um modelo de linguagem baseado na arquitetura Transformer
  - Modelo Bidirecional
  - Embeddings baseado em contexto
  - Utiliza a técnica de self-attention



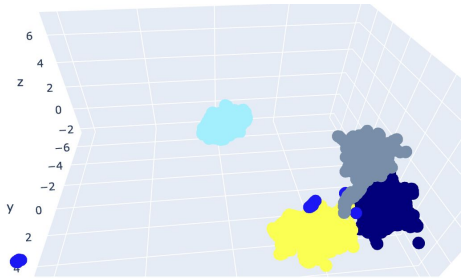


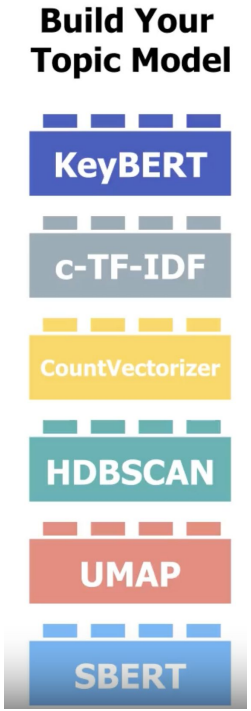
BERTopic

- UMAP é uma técnica utilizada para realizar redução de dimensionalidade
  - Neste caso, o UMAP é aplicado diretamente sobre os embeddings de sentenças gerados pelo Transformer

BERTopic

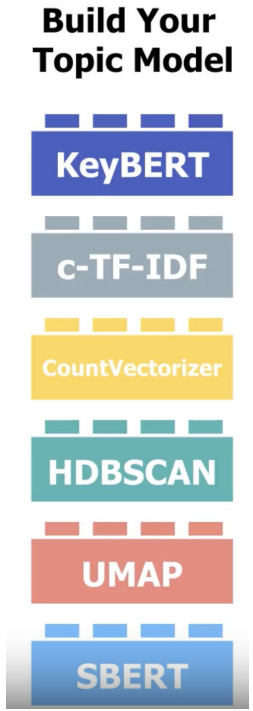
- HDBSCAN é um método de cluster hierárquico baseado em densidade
  - ➢ Utiliza os embeddings reduzidos para encontrar possíveis agrupamentos
  - ➢ Não precisa informar o número de agrupamentos





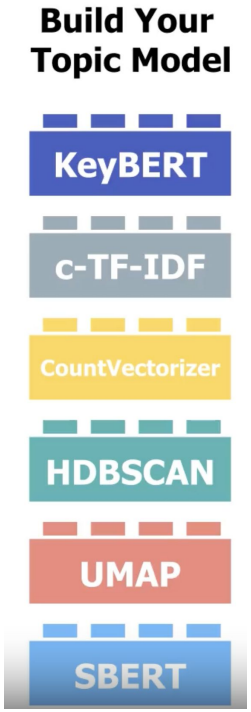
BERTopic

- CountVectorizer converte os documentos que estão agrupados em uma matriz com a frequência dos tokens/palavras
  - Lembrando que tópicos são formados por palavras
  - Com essa técnica, fazemos o mapeamento dos documentos em cada grupo e geramos a frequência de cada palavra



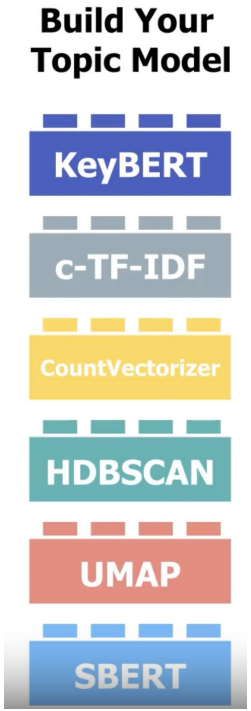
BERTopic

- c-TF-IDF é uma técnica de ponderação que foi adaptada para encontrar as palavras mais importantes entre os documentos de um mesmo agrupamento
  - Essas palavras mais importantes é o que formam um tópico!



BERTopic

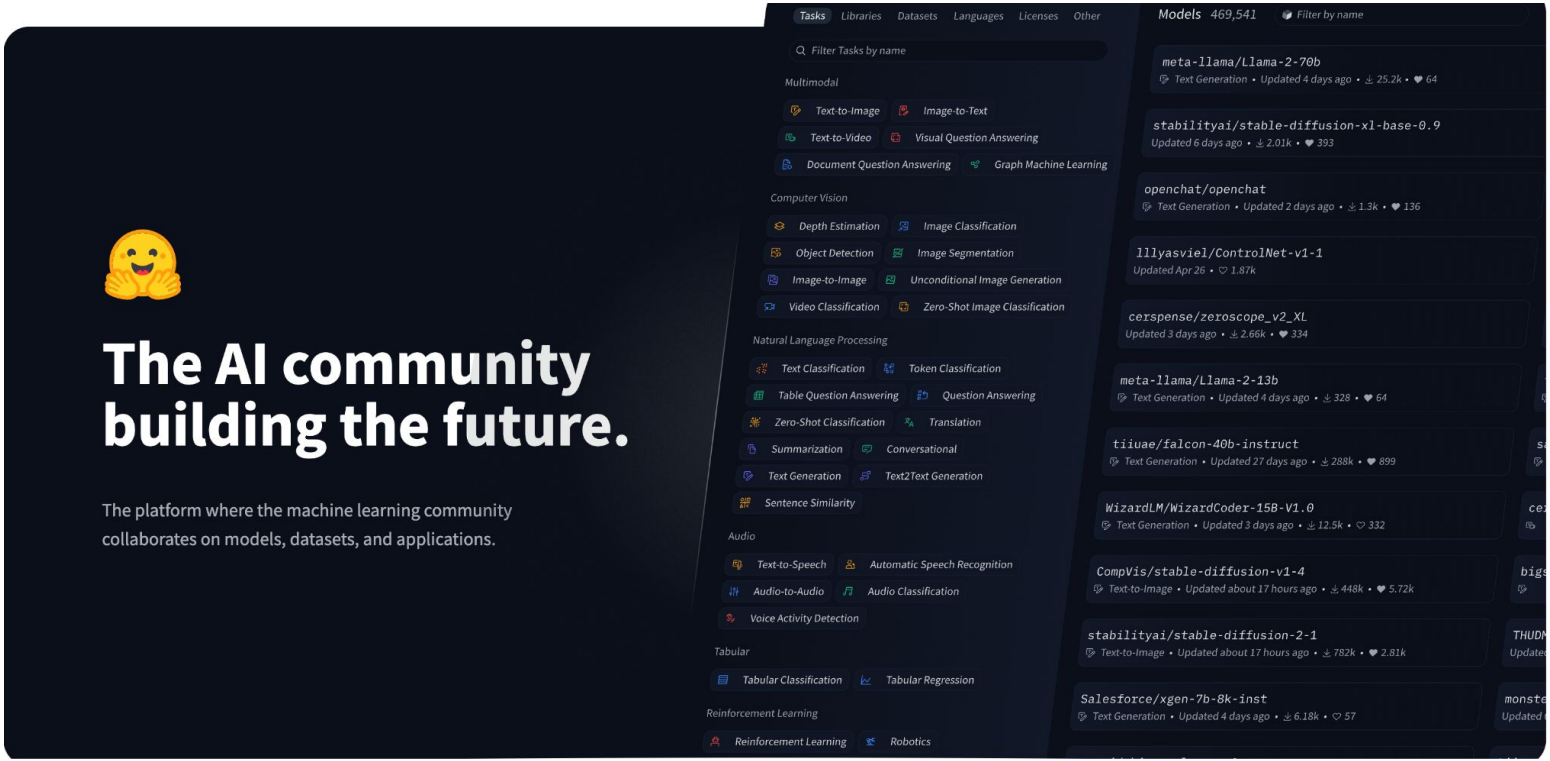
- É uma técnica para encontrar as palavras mais importantes em relação a um documento
  - A ideia é encontrar quais palavras possuem mais similaridade com relação ao documento
  - Para tal, usa-se word embedding e document embedding



5.
Hugging Face

The AI community building the future!

Hugging Face



Hugging Face

- Hugging Face é repositório de códigos/modelos de IA
- A empresa tem sido referência quando se trata de disponibilização de códigos/modelos de aprendizado profundo e que são open source
- Seu compromisso é com o desenvolvimento da IA livre (open)

Hugging Face

Usage

```
from transformers import AutoTokenizer  # Or BertTokenizer
from transformers import AutoModelForPreTraining  # Or BertForPreTraining for loa
from transformers import AutoModel  # or BertModel, for BERT without pretraining

model = AutoModelForPreTraining.from_pretrained('neuralmind/bert-base-portuguese-
tokenizer = AutoTokenizer.from_pretrained('neuralmind/bert-base-portuguese-cased'
```

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Universidade Federal Rural de Pernambuco - UAST

Minicurso: Topic Modeling using Transformer

Processamento de Linguagem Natural

Professor: Michael Oliveira da Cruz

UFRPE, 2023