



Documentação Técnica do *Text Comparator*

Data Science Team - Produto

Versão	Data	Autores	Último status
1.0	28/10/2022	João Souza	Criação do documento

Repositório do projeto

O repositório deste MVP pode ser acessado clicando no ícone ao lado: 

Ideia geral

Antes de apresentar os fluxogramas principais do MVP, é interessante introduzir a ideia macro deste MVP: comparar, **semanticamente**, dois textos. Existem, na literatura, diversas formas de comparação **estrutural** de textos, isto é, ao compará-los olhamos cada estrutura entre eles de forma *ipsis litteris*. Em algumas aplicações, essa pode não ser uma boa estratégia, uma vez que diferentes palavras podem ter o mesmo significado. Por exemplo, pense nas seguintes frases:

1. “Ele tem **diabetes** e foi lá tomar insulina”;
2. “Tomou insulina, pois é **diabético**”;
3. “Depois de desenvolver **TD2M**, começou a tomar insulina”.

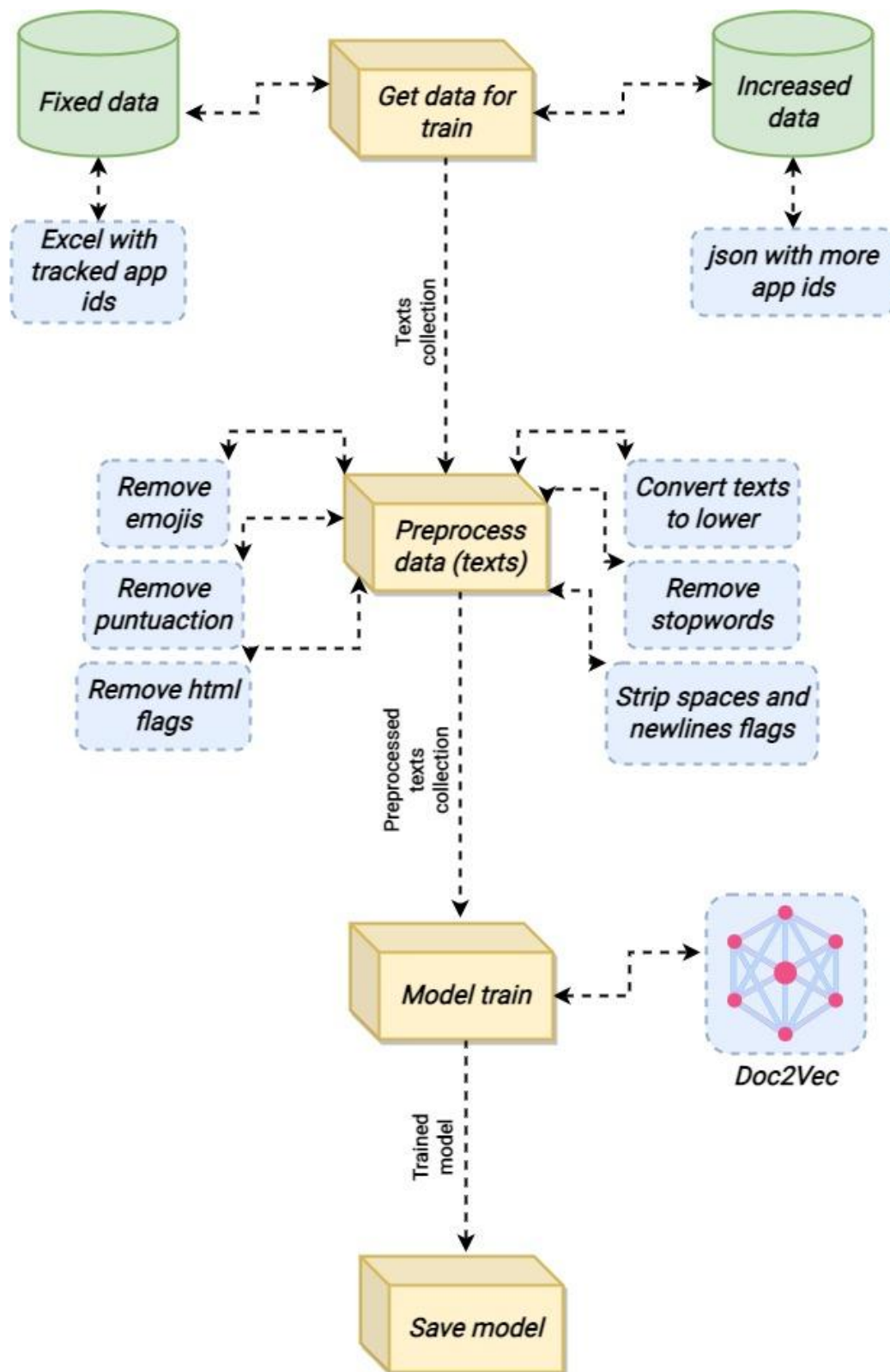
Semanticamente, possuem o mesmo sentido. Sintaticamente, não são iguais. Dessa forma, um comparador estrutural não veria a similaridade que o comparador semântico enxergaria.

Fluxogramas

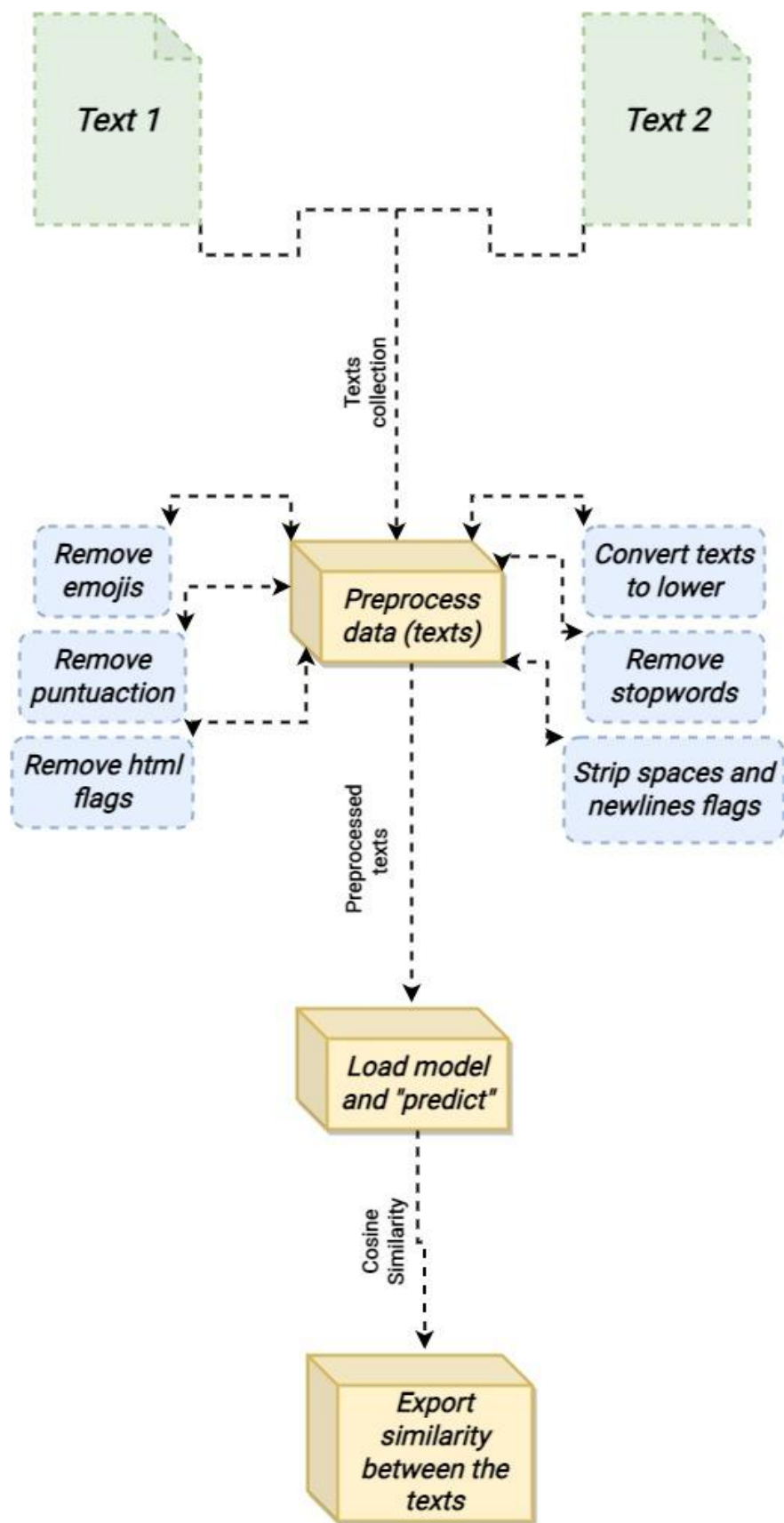
A seguir são apresentados os fluxogramas de execução de partes importantes do MVP, na premissa de ajudar a entender como o projeto está arquitetado.

Fluxograma de Treino

A estratégia de verificação de similaridade é baseada em um algoritmo de *Machine Learning* não supervisionado chamado *Doc2Vec*. Para realizar o treinamento do modelo, o seguinte fluxo foi adotado:



Fluxograma de Predição



Detalhes acerca do cálculo da similaridade

Como mostrado no fluxograma anterior, o primeiro passo para calcular a similaridade entre os textos é tratá-los de forma a deixar apenas palavras que trazem de fato informações úteis. A seguir mostraremos o formato de entrada e o que sai desta etapa:

Estrutura de *input*

Texto 1: “Então, quero que vocês entendam, que o melhor que a gente pode ter na vida, são as coisas básicas: é a nossa saúde, é a família, é um amigo, é um lugar pra viver, ta ligado?!... É ter no que acreditar, é viver em função de um sonho... Eu tenho uma alma, que é feita de sonhos !”

Texto 2: “Tão natural quanto a luz do dia Mas que preguiça boa, me deixa aqui à toa Hoje ninguém vai estragar meu dia Só vou gastar energia pra beijar sua boca Fica comigo então, não me abandona, não Alguém te perguntou como é que foi seu dia? Uma palavra amiga, uma notícia boa Isso faz falta no dia a dia A gente nunca sabe quem são essas pessoas Eu só queria te lembrar Que aquele tempo eu não podia fazer mais por nós Eu estava errado e você não tem que me perdoar Mas também quero te mostrar Que existe um lado bom nessa história Tudo que ainda temos a compartilhar E viver, e cantar Não importa qual seja o dia Vamos viver, vadiar O que importa é nossa alegria”

Array 1:

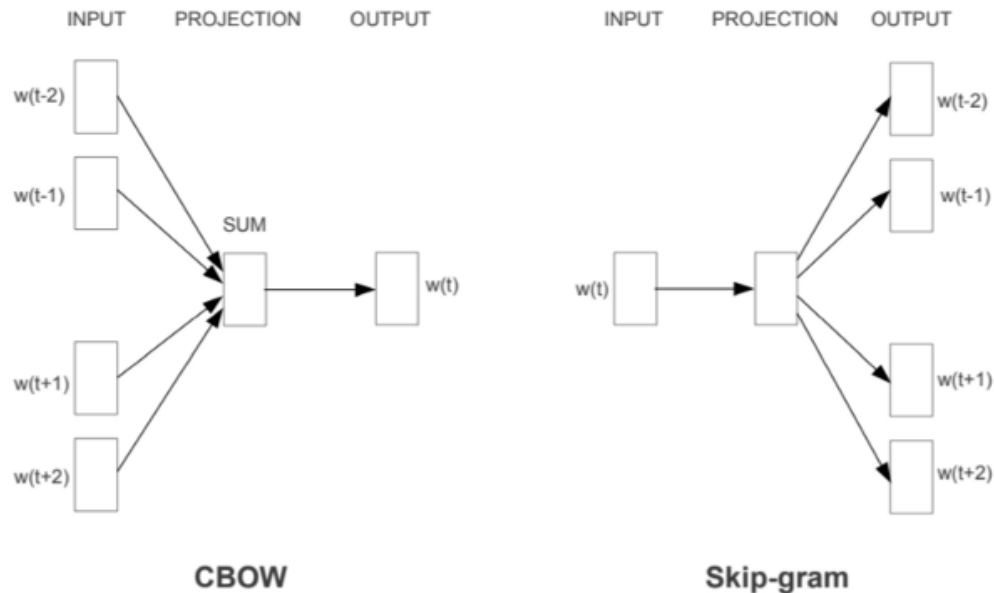
['então','quero','entendam','melhor','gente','pode','ter','vida','coisas','básicas','saúde','família','amigo','lugar','pra','viver','ta','ligado','ter','acreditar','viver','função','sonho','alma','feita','sonhos']

Array 2:

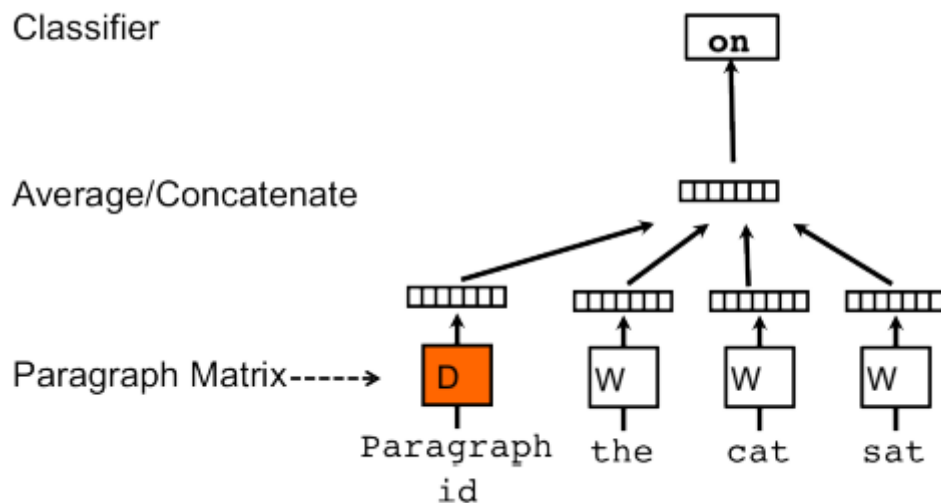
['tão','natural','quanto','luz','dia','preguiça','boa','deixa','aqui','toa','hoje','ninguém','vai','estragar','dia','vou','gastar','energia','pra','beijar','boca','fica','comigo','então','abandona','alguém','perguntou','dia','palavra','amiga','notícia','boa','faz','falta','dia','dia','gente','nunca','sabe','pessoas','queria','lembrar','tempo','podia','fazer','errado','perdoar','quero','mostrar','existe','lado','bom','nessa','história','tudo','ainda','compartilhar','viver','cantar','importa','dia','vamos','viver','vadiar','importa','alegria']

Doc2Vec

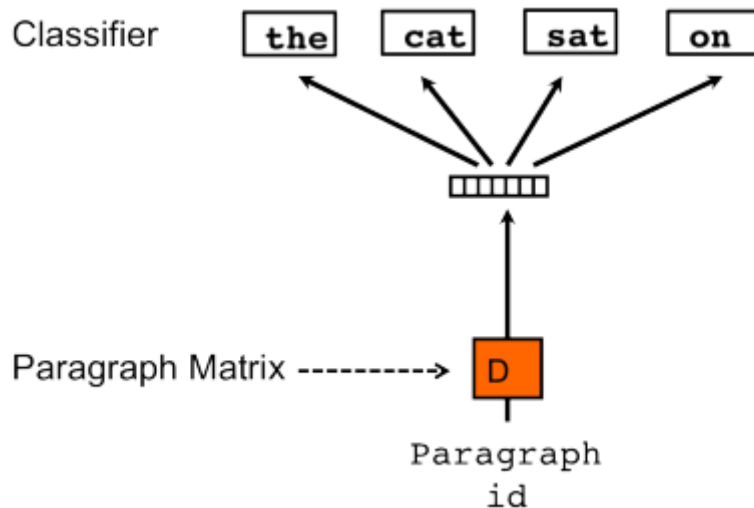
Após os *outputs* dos textos processados serem gerados, carregamos então o modelo *Doc2Vec* já treinando com uma base de descrições. O *Doc2Vec* opera através da **combinação** de duas estratégias: PV-DM (*Paragraph Vector - Distributed Memory*) e PV-DBOW (*Paragraph Vector - Distributed Bag of Words*).



Fonte: Medium - [Doc2Vec: Computing Similarity between Documents.](#)



Fonte: Medium - [Doc2Vec: Computing Similarity between Documents.](#)



Fonte: Medium - [Doc2Vec: Computing Similarity between Documents](#).

A abordagem da Rede Neural do tipo **PV-DBOW** funciona respondendo à seguinte pergunta: “dado um contexto de N palavras vizinhas a uma palavra W , qual será essa possível palavra W ?”

Já a abordagem **PV-DM**, funciona de maneira inversa: ela busca entender qual o contexto que uma palavra está inserida: “dada uma palavra W , quais são, em geral, as N palavras vizinhas dessa palavra W ?”

Com a combinação dessas duas abordagens, saímos apenas do aspecto estrutural do texto e conseguimos captar os aspectos contextuais de cada palavra ao longo dos textos.

Ao final da combinação, é gerado um *array* unidimensional que é a representação numérica dos textos através da utilização do *Doc2Vec* para cada texto de entrada.

Medindo a similaridade

Para medir a similaridade entre os dois textos, utilizamos atualmente a métrica da [Similaridade de Cossenos](#). Já que possuímos dois vetores numéricos, podemos calcular a similaridade como:

$$\text{similarity} = \cos \theta = \frac{A \cdot B}{||A|| ||B||}$$

onde A e B são os vetores numéricos gerados para os textos pelo *Doc2Vec*.

Logicamente, quando a similaridade é máxima é quando o $\cos \theta = 1$, ou seja, $\theta = 0^\circ$, dizendo que os dois vetores são idênticos. Dessa forma, à medida que o θ aumenta

em sentido anti-horário, a distância entre os dois vetores aumenta, diminuindo a similaridade entre os textos. Exemplo:

$$A = [1, 2, 3]$$

$$B = [1, 2, 3]$$

$$A \cdot B = (1 \cdot 1) + (2 \cdot 2) + (3 \cdot 3) = 14$$

$$\|A\| = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{1 + 4 + 9} = \sqrt{14}$$

$$\|B\| = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{1 + 4 + 9} = \sqrt{14}$$

$$\text{similarity} = \cos \theta = \frac{14}{\sqrt{14} \sqrt{14}}$$

$$\text{similarity} = \cos \theta = \frac{14}{14}$$

$$\text{similarity} = \cos \theta = 1$$