



**Data Science  
Academy**

[www.datascienceacademy.com.br](http://www.datascienceacademy.com.br)

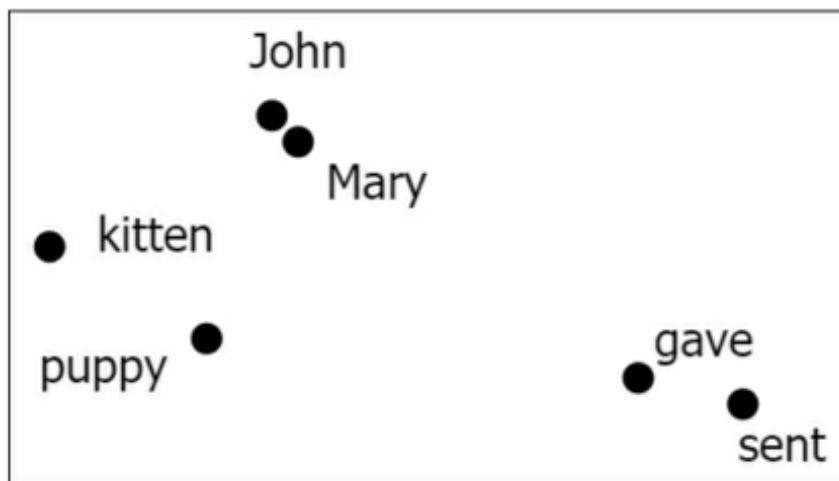
Processamento de Linguagem Natural

Usando Word Vectors com LSTMs

Outra maneira popular de melhorar o desempenho das LSTMs é usar vetores de palavras em vez de usar vetores codificados one-hot encoding como entrada para a LSTM. Vamos entender o valor desse método através de um exemplo. Vamos supor que queremos gerar texto a partir de alguma palavra aleatória. No nosso caso, seria o seguinte:

John \_\_\_\_\_.

Já treinamos nossa LSTM nas seguintes frases: John gave Mary a puppy. Mary has sent Bob a kitten. Vamos supor também que temos word vectors posicionados como mostrado na imagem abaixo:



As word embeddings dessas palavras, em sua forma numérica, podem se parecer com o seguinte:

- kitten: [0.5, 0.3, 0.2]
- puppy: [0.49, 0.31, 0.25]
- gave: [0.1, 0.8, 0.9]

Pode ser visto que  $\text{distance}(\text{kitten}, \text{puppy}) < \text{distance}(\text{kitten}, \text{gave})$ . No entanto, se usarmos one-hot encoding, elas serão as seguintes:

- kitten: [1, 0, 0, ...]
- puppy: [0, 1, 0, ...]
- gave: [0, 0, 1, ...]

Então,  $\text{distance}(\text{kitten}, \text{puppy}) = \text{distance}(\text{kitten}, \text{gave})$ .



Como já podemos ver, os vetores one-hot encoding não capturam a relação correta entre as palavras e veem todas as palavras com a mesma distância entre si. No entanto, vetores de palavras são capazes de capturar tais relacionamentos e são mais adequados como recursos em uma LSTM.

Usando vetores de palavras, a LSTM aprenderá a explorar melhor as relações entre as palavras. Por exemplo, com vetores de palavras, a LSTM aprenderá o seguinte: John gave Mary a kitten. Isso é bem parecido com o seguinte: John gave Mary a puppy. Mas, é bem diferente do seguinte: John gave Mary a gave. No entanto, este não seria o caso se vetores one-hot encoding fossem usados, pois as medidas de distância seriam as mesmas.