



# Assistente Virtual

WordVec x Transformers

**Grupo: Javali**

Bruno, Rafael, Caique, Marcus, Maria  
Gabriela, Mariana, Katiane e Silmara

**Disciplina**

PLN - Prof. Fabricio  
Galende

# ATIVIDADE

## G3. Tema: Assistente virtual utilizando Word2Vec

Desenvolva um assistente virtual, utilizando a modelagem Word2Vec, que efetue ações sobre uma interface de acordo com comandos efetuados pelo usuário. No seu projeto, considere o seguinte:

- a. Seu front end deverá possuir um prompt que aguarda comandos (não necessariamente estruturados, ou seja, podem ser diferentes das frases de treinamento do seu modelo) vindos do usuário.
- b. Os comandos devem incluir a modificação da cor de componentes da página, comandos de navegação (e.g.: navegar na barra de navegação da aplicação), o reposicionamento de componentes, aumento da fonte, etc.).
- c. Caso o assistente não reconheça o comando, o sistema deve notificar ao usuário que reformule o comando e, eventualmente, sugerir o comando tido como o mais “próximo” daquele efetuado pelo usuário.

Na sua modelagem de assistente, utilize obrigatoriamente um classificador do tipo Multilayer Perceptron (MLP) ou KNN.

# WORD2VEC

- Word2Vec é uma técnica que cria representações numéricas de palavras (word embeddings) usando redes neurais, organizando palavras em um espaço vetorial onde palavras com significados semelhantes ficam próximas.
- Baseada em uma tese popularizada por John Rupert Firth no fim da década de 1950. Esta tese diz que cada palavra é caracterizada, e até certo ponto definida, pelas outras palavras em seu entorno. Firth, por sua vez, baseou sua teoria na ideia de Zellig Sabbettai Harris, publicada em 1954, de que “palavras que ocorrem em contextos semelhantes tendem a ter significados semelhantes”.
- As duas principais arquiteturas são o Continuous Bag of Words (CBOW) e o Skip-gram, que são treinados em grandes corpora de texto para prever o contexto de uma palavra a partir dela mesma, ou o contrário.

# FUNCIONAMENTO

- O Word2Vec é treinado em grandes conjuntos de dados de texto (corpus) para aprender os pesos da rede neural.
- Após o treinamento, os pesos da rede se tornam os vetores de palavras (embeddings).
- As palavras são posicionadas no espaço vetorial de forma que palavras com contextos semelhantes (e, portanto, significados semelhantes) fiquem próximas umas das outras.

# CBOW

Este modelo é utilizado para descobrir a palavra central de uma sentença, baseado nas palavras que a cercam. Para representação, utilizaremos a seguinte notação:  
([palavras de contexto], palavra alvo)

## EXEMPLO:

O cachorro correu atrás do gato. Pode ser representada:

([O, correu], cachorro)

([cachorro, atrás], correu)

([correu, do], atrás)

([atrás, gato], do)

- Com estes dados, pretendemos “ensinar” ao modelo a predizer uma palavra alvo, baseada em palavras de contexto.

# SKIP-GRAM

Este é o segundo modelo utilizado no Word2Vec, porém aqui ao invés de tentar descobrir a palavra central, é feito o processo inverso: da palavra central, tentamos descobrir as palavras de contexto.

## EXEMPLO:

O cachorro correu atrás do gato , e a notação (palavra central, [palavras de contexto] .

- 1.(cachorro, [O, correu])
- 2.(correu, [cachorro, atrás])
- 3.(atrás, [correu, do])
- 4.(do, [atrás, gato])

# TRANSFORMERS

- É uma arquitetura de rede neural que revolucionou a área de Inteligência Artificial (IA), especialmente o Processamento de Linguagem Natural (PLN). Ela permite que modelos de IA entendam o contexto e o significado de dados sequenciais, como palavras em uma frase, rastreando relacionamentos entre eles.
- Principal diferencial: Permite que o modelo atribua pesos diferentes a diferentes partes do texto de entrada ao processar uma palavra específica, independentemente da distância entre elas na sequência. Isso ajuda a capturar dependências de longo prazo e entender o contexto global da informação.

# FUNCIONAMENTO

- Se baseia no mecanismo de autoatenção (self-attention) e no processamento paralelo, superando as limitações das redes neurais recorrentes (RNNs).
- Em resumo, a tecnologia Transformer funciona aprendendo o contexto e o significado dos dados sequenciais (como texto ou fala) por meio do rastreamento de relações e dependências entre os elementos, independentemente de sua posição na sequência, e processando-os de forma altamente paralela.

# BERT

- O BERT é um modelo que se baseia especificamente no codificador (encoder) da arquitetura Transformer original e o utiliza de maneira bidirecional. Isso significa que ele processa o texto olhando para a esquerda e para a direita de cada palavra simultaneamente para entender o contexto completo de uma frase.
- Ao contrário de modelos anteriores que liam o texto em apenas uma direção (esquerda para direita, ou vice-versa)

# COMO FUNCIONA

Durante o pré-treinamento, o BERT utiliza duas estratégias principais:

1. Masked Language Modeling (MLM): Cerca de 15% das palavras em uma frase são aleatoriamente ocultadas e o modelo tenta prever quais são essas palavras com base no contexto ao redor delas.
2. Previsão da Próxima Sentença (Next Sentence Prediction - NSP): O modelo é alimentado com pares de frases e precisa prever se a segunda frase é a continuação lógica da primeira.

# APRESENTAÇÃO PRÁTICA

OBRIGADO PELA  
ATENÇÃO