UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO

Processamento de Linguagem Natural
Professor: Thales Vieira

3a lista de exercícios **2 de setembro de 2025**

Instruções:

A lista deve ser respondida por grupos de até 2 pessoas (graduação) e individualmente (mestrado).
Resoluções idênticas de grupos distintos serão desconsideradas.
O código e demais dados devem ser anexados a cada questão.
Data limite para entrega: 18/09/2025.

Usando sua base de textos após os pré-processamento realizados na lista 1, realize as seguintes tarefas:

1. O objetivo dessa questão é desenvolver buscadores de palavras e documentos.

a) escolha e aplique um modelo do tipo word2vec **pretreinado** a seus textos, compatível com o idioma dos textos (inglês ou português).
b) escolha 5 palavras de consulta que não estejam em nenhum dos textos. Para cada palavra de consulta, encontre as 3 palavras **de seu conjunto de textos** mais parecidas com cada uma das palavras de consulta e exiba os documentos onde estas palavras aparecem.
c) Seja d um documento da base e w uma palavra de consulta. Implemente o seguinte algoritmo para buscar documentos:
    1. Encontre $L_5(d, w)$: a lista com as 5 palavras mais parecidas com w em um certo documento d.
    2. Para cada documento d, calcule a distância média $DM_5(d, w)$ entre w e as palavras de $L_5(d, w)$.
    3. Recupere os 3 documentos da base que tenham a menor distância média $DM_{10}(d, w)$.
d) aplique o algoritmo da letra c) para buscar documentos a partir de 5 palavras distintas de consulta, e exiba os 3 documentos mais próximos de cada uma.

2. Resolva novamente a primeira questão da 2a lista e compare com os resultados obtidos anteriormente:

a) Aplicando a representação vetorial Doc2Vec combinado com os classificadores usados anteriormente.
b) Usando uma arquitetura de rede neural que utilize camadas de Embedding e LSTM. Aqui você deve usar como embedding vetores obtidos a partir do modelo word2vec adotado na primeira questão.

3. Usando sua base de textos e a biblioteca spaCy, realize as seguintes tarefas:

a) Extraia as etiquetas gramaticais (POS) de cada token do seu textos.
b) Calcule e plote um gráfico com as frequências de cada tipo gramatical.
c) Escolha um tipo de entidade nomeada pretreinada em algum modelo da spaCy. Dê preferência a um tipo de entidade que tenha uma boa chance de aparecer mais de uma vez em cada documento. Realize o reconhecimento destas entidades em seus textos.
d) Gere um grafo com pesos onde os nós representam cada entidade reconhecida, e as arestas são criadas caso pares de entidades apareçam juntas num mesmo documento. O peso das arestas deve ser igual à quantidade destas ocorrências. Plote o grafo usando a biblioteca NetworkX, onde as arestas devem ser desenhadas com espessura proporcional ao peso.

4. Estude o tutorial *Character-level recurrent sequence-to-sequence model* disponível em `https://keras.io/examples/nlp/lstm_seq2seq/`.

a) Treine um outro modelo de tradução entre línguas distintas. Você pode encontrar conjuntos de treinamento em `http://www.manythings.org/anki/`. Exiba 5 exemplos de tradução de frases curtas.
b) Adapte a arquitetura para operar a nível de palavras (e não caracteres), usando uma camada do tipo *Embedding*. Exiba 5 exemplos e avalie qual funciona melhor de forma qualitativa.