Job Description Similarity Detection

An end-to-end Natural Language Processing (NLP) system to identify and compare job descriptions by semantic similarity.

This project leverages both Python and C++ for high-performance data cleaning, feature extraction, and cosine similarity detection.

Project Summary

In this project, we designed a full NLP pipeline to analyze, clean, vectorize, and compute similarity between job descriptions.

Motivation:

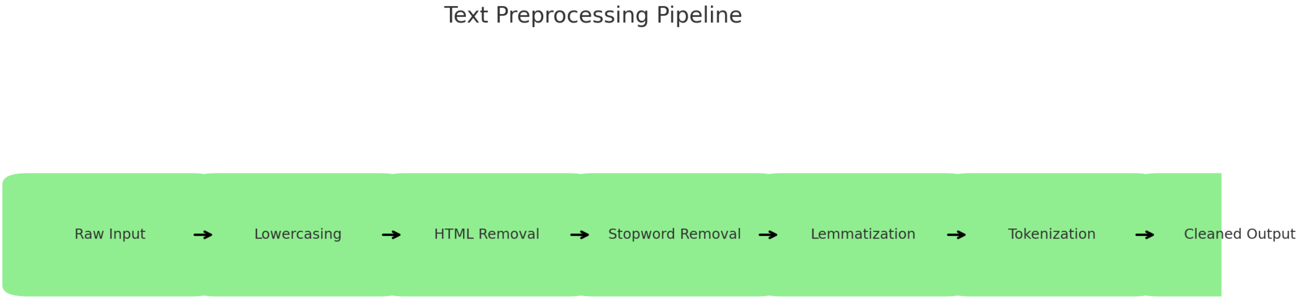
- Job postings are often duplicated or paraphrased versions of the same role.
- HR systems benefit from automated similarity detection to improve recommendations and avoid duplication.

Tools:

- Python for data manipulation and evaluation
- C++ for fast similarity detection
- TF-IDF & Bag-of-Words for feature extraction
- Spacy, NLTK, TextBlob for preprocessing variants

Job Description Similarity - NLP Project (C++ & Python)

NLP Workflow Pipeline



Overview of the text processing pipeline from raw job descriptions to vectorized formats.

Function Overview

Preprocessing

remove_noise_regex/html: Clean HTML tags and noisy characters.

remove_numbers, punctuation: Clean digits and punctuation.

lemmatize_*: Convert words to root form.

stem_*: Simplify words to their core stem.

remove_stopwords_*: Eliminate uninformative common words.

Feature Extraction

TF-IDF: Numeric vectors reflecting term importance.

Bag-of-Words: Simpler frequency-based representation.

Similarity Metrics

cosine_similarity: Compares document vectors using angle distance.

jaccard_similarity: Measures token-set overlap.

edit_distance: Character-level distance calculation.

Example Similarity Scores

```
Similarity(doc 0 vs doc 1) = 0.53
Similarity(doc 0 vs doc 2) = 0.07
Similarity(doc 1 vs doc 3) = 0.64
Similarity(doc 2 vs doc 3) = 0.08
```

Scores close to 1 indicate higher similarity between job descriptions.

Future Enhancements

- Add advanced semantic models like Word2Vec and BERT

- Extend for multilingual support

- Deploy as an interactive web API

- Visualize document clusters using PCA or t-SNE