Criando o arquivo geracao\_resumo.py para Geração de Resumos de Vídeos

**Entendendo a Geração de Resumos:**

A geração de resumos de vídeos consiste em criar uma descrição concisa e informativa do conteúdo principal do vídeo. Isso envolve a extração das informações mais relevantes, como as atividades principais, as emoções dominantes e os eventos-chave.

**Utilizando Técnicas de Processamento de Linguagem Natural (NLP):**

Para gerar resumos, podemos utilizar técnicas de NLP como:

* **Extração de frases-chave:** Identificar as palavras e frases mais importantes no texto.
* **Sumarização extractiva:** Selecionar as frases mais relevantes do texto original para formar o resumo.
* **Sumarização abstractiva:** Gerar novas frases que capturem o significado do texto original, utilizando modelos de linguagem.

**Código para o arquivo geracao\_resumo.py:**

Python

import nltk  
from nltk.tokenize import sent\_tokenize  
from nltk.corpus import stopwords  
from nltk.cluster.util import cosine\_distance  
import numpy as np  
  
def gerar\_resumo(texto):  
 """Gera um resumo do texto utilizando a técnica de sumarização extractiva.  
  
 Args:  
 texto (str): Texto a ser resumido.  
  
 Returns:  
 str: Resumo do texto.  
 """  
  
 # Pré-processamento  
 stop\_words = set(stopwords.words('portuguese'))  
 sentences = sent\_tokenize(texto)  
 word\_embeddings = {}  
  
 # Criar uma matriz de similaridade entre as frases  
 for i, sent in enumerate(sentences):  
 words = nltk.word\_tokenize(sent)  
 words = [w for w in words if not w in stop\_words]  
 word\_embeddings[i] = words  
  
 similarity\_matrix = np.zeros((len(sentences), len(sentences)))  
 for i in range(len(sentences)):  
 for j in range(len(sentences)):  
 if i != j:  
 similarity\_matrix[i][j] = cosine\_distance(word\_embeddings[i], word\_embeddings[j])  
  
 # Selecionar as frases mais importantes  
 sentence\_similarity\_graph = np.array(similarity\_matrix)  
 sentence\_similarity\_graph[sentence\_similarity\_graph < (1 - (1/len(sentence\_similarity\_graph))) ] = 0  
 sentence\_indices = np.argsort(-np.sum(sentence\_similarity\_graph, axis=0))[:len(sentences) // 2]  
 summary = ' '.join([sentences[i] for i in sentence\_indices])  
 return summary  
  
# Exemplo de uso:  
if \_\_name\_\_ == "\_\_main\_\_":  
 texto = "Este é um exemplo de texto. Ele será utilizado para gerar um resumo. O resumo deve ser conciso e informativo."  
 resumo = gerar\_resumo(texto)  
 print(resumo)

**Explicação do Código:**

* **Pré-processamento:** O texto é dividido em frases e as stop words são removidas.
* **Criação da matriz de similaridade:** Uma matriz é criada para calcular a similaridade entre cada par de frases.
* **Seleção das frases:** As frases mais importantes são selecionadas com base na similaridade e utilizadas para gerar o resumo.

**Observações:**

* **Sumarização extractiva:** Este código implementa uma técnica de sumarização extractiva, que seleciona as frases mais relevantes do texto original.
* **Limitações:** A qualidade do resumo depende da qualidade do texto original e da escolha dos parâmetros.
* **Modelos de linguagem:** Para gerar resumos mais sofisticados, você pode utilizar modelos de linguagem como o BERT ou o GPT-3.
* **Adaptação para vídeos:** Para gerar resumos de vídeos, você precisará converter a informação extraída do vídeo (rostos, emoções, atividades) em texto e aplicar as técnicas de sumarização.

**Próximos Passos:**

* **Integração com a análise de vídeo:** Conectar a geração de resumos com as etapas anteriores, utilizando as informações sobre rostos, emoções e atividades para gerar um texto descritivo.
* **Personalização:** Adaptar o algoritmo de sumarização para diferentes tipos de vídeos e requisitos.
* **Avaliação:** Avaliar a qualidade dos resumos gerados utilizando métricas como ROUGE.

**Considerações:**

* **Complexidade:** A geração de resumos de vídeos é uma tarefa desafiadora que envolve diversas áreas, como visão computacional, processamento de linguagem natural e aprendizado de máquina.
* **Contexto:** O contexto do vídeo é importante para gerar resumos mais precisos.
* **Multimodalidade:** A combinação de informações visuais, textuais e de áudio pode melhorar a qualidade dos resumos.

**Gostaria de adicionar alguma funcionalidade específica à sua geração de resumos?**

**Por exemplo:**

* **Resumos personalizados:** Gerar resumos adaptados a diferentes públicos ou interesses.
* **Resumos multilíngues:** Gerar resumos em diferentes idiomas.
* **Resumos interativos:** Permitir ao usuário explorar o vídeo de forma interativa, com base no resumo gerado.

**Lembre-se de adaptar este código às suas necessidades específicas e à estrutura do seu projeto.**

**Fontes**

1. <https://github.com/AarushMuthukrishnan/pamedia-2023>

2. <https://wenku.csdn.net/answer/4aa23efa3c214b839da49eb1769a75a3>