Pesquisa de Similaridade de Jogos por Gênero - IGDB Games Dataset

Diego de Farias Carazo Prieto

3 de Novembro de 2025

Sumário

1 Introdução e Objetivo

A análise de grandes volumes de dados textuais para identificar relevância temática é um desafio central no Processamento de Linguagem Natural (PLN). Este projeto se insere nesse contexto, buscando explorar as similaridades conceituais em um vasto repositório de metadados de jogos.

1.1 Descrição do Dataset

O **IGDB Games Dataset** é um banco de dados sobre videogames, tendo uma extensa biblioteca de informações detalhadas sobre títulos, gêneros, plataformas e data de lançamento dos jogos. O arquivo original, disponibilizado em CSV, contém 380179 linhas e 60 colunas.

Esta pesquisa utiliza primariamente as seguintes colunas para ranqueamento e análise:

- **Name (Título):** O título do jogo.
- **Summary (Sinopse):** O resumo/sinopse do jogo, utilizado como *corpus* textual.

1.2 Objetivo Central da Pesquisa

O objetivo primordial desta pesquisa é demonstrar a eficácia do modelo **TF-IDF** na recuperação de informações temáticas específicas, aplicando-o contra a *query* semântica: **"dark gothic vampire world"**.

Ao vetorizar a sinopse de cada jogo e a *query*, é calculada a **Similaridade do Cosseno** para ranquear os jogos que apresentam maior afinidade temática com os conceitos de escuridão, estética gótica e a presença de vampirismo.

2 Metodologia de Similaridade

2.1 Metodologia: TF-IDF e Similaridade do Cosseno

Para mensurar a relevância temática, o projeto emprega a técnica estatística **TF-IDF** (do inglês, *Term Frequency–Inverse Document Frequency*), combinada com a Similaridade do Cosseno.

> O **TF-IDF** é uma métrica que visa quantificar a importância de um termo para um documento específico dentro de uma coleção (o *corpus*). Ele faz isso ponderando a frequência de uma palavra no documento (*Term Frequency*) pela raridade dessa palavra em toda a coleção de documentos (*Inverse Document Frequency*). Palavras-chave raras e específicas (como *vampire* ou *gothic*) recebem um peso maior do que palavras comuns, permitindo que o modelo identifique o conteúdo único de cada sinopse.

2.2 Metodologia de Implementação

O processo de similaridade textual foi dividido em quatro etapas principais, conforme detalhado nas subseções a seguir.

2.3 Visualização do Código Implementado

O processo de similaridade foi dividido logicamente em cinco blocos principais no código Python, garantindo a rastreabilidade das etapas de pré-processamento, vetorização e ranqueamento.

2.3.1 Pré-Processamento dos Dados

Esta etapa é crucial para garantir que apenas os termos mais relevantes contribuam para o cálculo de similaridade e evitar ruído textual.

1. **Carregamento e Seleção:** O *dataset* foi carregado através da biblioteca `pandas`. O projeto focou em três colunas principais: `name` (Nome do Jogo), `summary` (Sinopse) e `tags` (Tags).

2. **Limpeza de Texto:** Uma função de limpeza (`clean_text`) foi aplicada à coluna de sinopses (`summary`), realizando as seguintes transformações:

```python
stop_words = set(stopwords.words(IDIOMA_STOPWORDS))

def clean_text(text):
    """Realiza a limpeza básica: minúsculas, remove pontuação e stopwords."""
    if pd.isna(text) or text is None:
        return ""

    text = str(text).lower()

    text = re.sub(r'[^a-z\s]', '', text)

    word_tokens = word_tokenize(text)

    filtered_words = [w for w in word_tokens if w not in stop_words and len(w) > 1]

    return " ".join(filtered_words)
```

Figura 1 – Função Python para limpeza de texto (`clean_text`), incluindo remoção de stopwords.

- Conversão para minúsculas.
- Remoção de pontuações e caracteres especiais (*regex* `[a-z\s]`).
- Tokenização das sentenças (`nltk.word_tokenize`).
- Remoção de *stopwords* (palavras irrelevantes comuns, como 'a', 'o', 'de', 'e') definidas para o idioma Inglês.
- Filtro final para remover palavras com apenas uma letra.

3. **Preparo da *Query*:** O termo de busca (`"dark gothic vampire world"`) foi submetido ao mesmo processo de pré-processamento para que estivesse no mesmo formato dos dados do *dataset*.

```python
try:
    df = pd.read_csv(NOME_ARQUIVO, sep=';')
    print(f"Arquivo '{NOME_ARQUIVO}' carregado com sucesso.")
except FileNotFoundError:
    print(f"Erro: O arquivo '{NOME_ARQUIVO}' não foi encontrado.")
    exit()


colunas_necessarias = [COLUNA_TEXTO, COLUNA_NOME_JOGO, COLUNA_SUMMARY]
for col in colunas_necessarias:
    if col not in df.columns:
        print(f"Erro: Coluna obrigatória '{col}' não encontrada. Colunas disponíveis: {list(df.columns)}")
        exit()

df['cleaned_tags'] = df[COLUNA_TEXTO].apply(clean_text)
```

Figura 2 – Carregamento do dataset, validação das colunas e aplicação inicial da função de limpeza.

2.3.2 Geração da Matriz TF-IDF

O **TfidfVectorizer** da biblioteca `sklearn` foi configurado e aplicado aos dados limpos.

- **Inicialização do Vetorizador:** O vetorizador foi configurado para gerar *unigrams* e *bigrams* (`ngram_range=(1, 2)`).
- **Filtros Estatísticos:** Foram aplicados filtros para otimizar o vocabulário, ignorando termos que aparecem em mais de 85% dos documentos (`max_df=0.85`) e termos que aparecem em menos de 5 documentos (`min_df=5`).
- **Ajuste e Transformação:** O vetorizador foi treinado nos dados limpos (`df['cleaned_tags']`) para gerar a matriz esparsa TF-IDF.

```
query_cleaned = clean_text(TERMOS_DE_BUSCA)
query_vector = vectorizer.transform([query_cleaned])

similarity_scores = tfidf_matrix.dot(query_vector.transpose()).toarray().flatten()

df['goth_vampire_score'] = similarity_scores
df_sorted = df.sort_values(by='goth_vampire_score', ascending=False)
tfidf_matrix = vectorizer.fit_transform(df['cleaned_tags'])
print(f"Matriz TF-IDF criada com {tfidf_matrix.shape[0]} itens e {tfidf_matrix.shape[1]} termos/features.")
```

Figura 3 – Inicialização e treinamento do `TfidfVectorizer` (`sklearn`) e criação da matriz TF-IDF.

2.3.3 Cálculo da Similaridade de Cosseno

Para ranquear os jogos, foi utilizada a Similaridade do Cosseno.

- **Vetorização da *Query*:** O vetorizador treinado foi usado para transformar a *query* limpa (`query_vector`) no espaço vetorial dos documentos.
- **Cálculo da Similaridade:** A similaridade foi calculada por meio do **produto escalar** da matriz TF-IDF pelo vetor da *query*.
- **Pontuação (SCORE):** O resultado desse cálculo gerou um *score* de similaridade (`goth_vampire_score`) para cada jogo.

2.3.4 Ranqueamento e Exibição dos Resultados

Na etapa final, o *score* de similaridade foi adicionado ao *DataFrame* original e classificado em ordem decrescente.

- O *DataFrame* foi classificado em ordem decrescente de *score* (`df_sorted = df.sort_values(by ascending=False)`).

```
query_cleaned = clean_text(TERMOS_DE_BUSCA)
query_vector = vectorizer.transform([query_cleaned])

similarity_scores = tfidf_matrix.dot(query_vector.transpose()).toarray().flatten()

df['goth_vampire_score'] = similarity_scores
df_sorted = df.sort_values(by='goth_vampire_score', ascending=False)
```

Figura 4 – Cálculo da Similaridade do Cosseno via produto escalar entre a matriz TF-IDF e o vetor da query.

- Os **10 jogos mais relevantes** foram selecionados e formatados para exibição.

```
print("\n" + "="*80)
print(f"TOP {NUM_PRINCIPAIS_ITENS} JOGOS MAIS RELEVANTES PARA O TEMA: '{TERMOS_DE_BUSCA}' (Baseado nas Tags)")
print("="*80)

for i in range(min(NUM_PRINCIPAIS_ITENS, len(df_sorted))):
    score = df_sorted.iloc[i]['goth_vampire_score']
    game_name = df_sorted.iloc[i][COLUNA_NOME_JOGO]
    summary = df_sorted.iloc[i][COLUNA_SUMMARY]
    original_tags = df_sorted.iloc[i][COLUNA_TEXTO]

    summary_display = str(summary) if pd.notna(summary) else "N/A"

    print(f"\n✨ RANK {i+1} | SCORE: {score:.4f} ✨")
    print("-" * 30)
    print(f"**NOME DO JOGO:** {game_name}")

    print(f"**TAGS:** {summary_display[:150]}...")

    print(f"**RESUMO:** {original_tags}")
    print("-" * 30)

print("\n" + "="*80)
```

Figura 5 – Loop final de ranqueamento, extração dos dados e formatação de saída dos 10 jogos mais relevantes.

3 Resultados e Análise

A Tabela 1 apresenta os 10 jogos com maior similaridade temática em relação à *query* "dark gothic vampire world", juntamente com suas pontuações.

Tabela 1 – Top 10 Jogos Mais Relevantes para o Tema

RANK	NOME DO JOGO	SIMILARIDADE
1	Dark Legends	0.4277
2	Warhammer 40,000: Storm Of Vengeance	0.3920
3	Lady Hunt	0.3325
4	Tales From The Under-Realm: After Midnight	0.3311
5	Demoniaca: Everlasting Night	0.3173
6	Elminage Gothic	0.3118
7	Vampire Ventures	0.2983
8	Blade	0.2889
9	Hell Of A Marriage	0.2834
10	Immortal Realms: Vampire Wars	0.2826

1. **Dark Legends** (Similaridade: 0.4277)
   Resumo: Vampire game.

2. **Warhammer 40,000: Storm Of Vengeance** (Similaridade: 0.3920)
   Resumo: Warhammer 40,000: Storm Of Vengeance is a lane strategy game set in the dark, gothic universe of Warhammer 40,000.

3. **Lady Hunt** (Similaridade: 0.3325)
   Resumo: Lady Hunt is a pixel art action platformer, set in a dark gothic world, where blood has been infected. You play as a huntress, who must find a way to purify this world, inhabited by monsters and beasts.

4. **Tales From The Under-Realm: After Midnight** (Similaridade: 0.3311)
   Resumo: TFTU: After Midnight is a dark / gothic visual novel with multiple endings. Your choices matter and will determine who stays alive and who dies. Discovering the whole truth about the murders will require more than one playthrough.

5. **Demoniaca: Everlasting Night** (Similaridade: 0.3173)
   Resumo: Demoniaca - is a dark, gothic, mature and sexy action rpg inspired by the Castlevania series. Explore, fight, meet new friends and avenge those who brought you unbearable pain and loss.

6. **Elminage Gothic** (Similaridade: 0.3118)
   Resumo: From the makers of the world renowned “Wizardry Empire” titles, Starfish

SD brings you the latest entry in their popular series of dungeon crawlers. Elminage Gothic, previously only available in Japanese on PlayStation Portable, now comes to PC offering a classic old school dungeon crawling experience with a dark, gothic twist!

7. **Vampire Ventures** (Similaridade: 0.2983)
   Resumo: A match-3 quest in which Vampire Valory has to find her father who was kidnapped by Vladoric the Vampire.

8. **Blade** (Similaridade: 0.2889)
   Resumo: You're blade, Gothic City's baddest vampire hunter. When rival vampire clams ignite a deadly feud, you go deep to do some serous demon killing. Stalk the undead through 24 gothic locations, then waste an arsenal of intense weapons. Plan on staying up late tonight 'cause and even greater evil awaits.

9. **Hell Of A Marriage** (Similaridade: 0.2834)
   Resumo: Faustian Gothic Gay Romance

10. **Immortal Realms: Vampire Wars** (Similaridade: 0.2826)
    Resumo: Immortal Realms: Vampire Wars is an engaging strategy game set in a dark vampire world in turmoil, that combines empire management and turn-based combat with unique card-game elements. Descend into a mythical world filled with horrors and legends – and hurl yourself into a compelling gothic adventure paired with a challenging game experience. Discover the secrets of Nemire and experience an enthralling story from the perspective of four mighty vampire lords, each with their own goals and agendas.

3.1 Análise dos Ângulos Comuns

Para realizar a Análise de Ângulos Comuns, foram identificados os termos temáticos da *query* — **Vampiro**, **Gótico/Dark** e **Mundo/Cenário** — nos resumos dos 10 jogos listados.

1. **Prevalência do Ângulo Gótico/Dark:** Este é o ângulo mais presente, aparecendo explicitamente em **9 dos 10** resumos, destacando a forte correlação entre os adjetivos e o ranqueamento. O caso de *Warhammer 40,000: Storm Of Vengeance* (Rank 2) é notável, pois sua alta pontuação é devida quase integralmente à presença de "dark, gothic universe"na sinopse, mesmo sem tags listadas, indicando o peso do texto para o modelo.

2. **O Ângulo Vampiro:** O tema **Vampiro** ou caçador de vampiros aparece em **5 dos 10** jogos. A alta pontuação de *Dark Legends* (Rank 1) sugere que a palavra

"Vampire"possui um peso semântico altíssimo na função de similaridade. Os jogos que combinam Vampiro e Gótico (ex: *Immortal Realms*, *Blade*) tendem a estar na faixa de similaridade média-baixa (0.28-0.29).

3. **O Ângulo Mundo/Cenário:** Este ângulo, expresso por termos como "world"ou "universe", é o menos comum, aparecendo explicitamente em apenas **4 dos 10** jogos. Isso indica que, embora seja um termo da *query*, ele foi o menos determinante para a pontuação final de similaridade.

4 Conclusão

A pesquisa demonstra que os jogos mais semelhantes à *query* possuem uma forte e direta correspondência com os termos centrais da busca, validando a eficácia do método TF-IDF na mineração de texto para identificação de similaridade temática em grandes *datasets*. Embora todos os termos não tenham sido encontrados em conjunto, mesmo com as temáticas tendo uma relação, demonstra que estes jogos correspondem à pesquisa, e estão relacionados, mesmo sem menção.