PROJETO GENOME

Contexto e Desafio

- Ao longo do curso, utilizando os dados fornecidos pela Startup Genome, um dataset de 2GB com mais de 9 milhões de linhas, foi possível detectar o quanto a classificação de patentes exige expertise e é um processo caro e demorado por conta do alto volume de dados envolvido.

- De acordo com o *report* da empresa *SkyQuest Technology*, o mercado de patentes está em crescimento: aumento de 10,1% nas concessões em 2023.O mesmo *report* trouxe a estimativa de que tal mercado é avaliado em U$1.12 Bi e pode chegar a U$3.39 Bi até 2032.

- Ao investigar os reports existentes no mercado, percebemos que são dados estáticos (muitas vezes sobre o ano anterior), caros e limitados a poucos players. Com isso, notamos a lacuna no mercado, para uma solução interativa e dinâmica para insights em tempo real.

Solução Proposta

Patent Insight Hub

- Plataforma para classificação e análise automática de patentes através do uso de *deep learning* e PLN (***BERT/SciBERT*** + modelos supervisionados)

- Utilização de IA (RAG + GPT-4) para justificar classificações e tornar o processo mais didático ao usuário.

- Dashboards dinâmicos para insights em tempo real a respeito de setores específicos, ecossistemas emergentes, entre outros pontos de análise.

Visão do MVP

```
# 3. Gerar embeddings com BERT

tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertModel.from_pretrained('bert-base-uncased')

def gerar_embedding(text):
    inputs = tokenizer(text, return_tensors='pt', truncation=True, padding=True, max_length=512)
    outputs = model(**inputs)
    return outputs.last_hidden_state.mean(dim=1).squeeze().detach().numpy()

df_patents['bert_embedding'] = df_patents['processed_abstract'].apply(gerar_embedding)

df_subsetores['bert_embedding'] = df_subsetores['KeyWords'].apply(lambda keywords: gerar_embedding(' '.join(str(keywords).split(','))))
```

```
# 4. Preparar dados para classificação

X = list(df_patents['bert_embedding'])

# y = df_patents['predicted_subsetor'].astype('category').cat.codes
y = df_patents['processed_abstract'].astype('category').cat.codes

y = to_categorical(y)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Solução Proposta

Patent Insight Hub

- Plataforma para classificação e análise automática de patentes através do uso de *deep learning* e PLN (***BERT/SciBERT*** + modelos supervisionados)

- Utilização de IA (RAG + GPT-4) para justificar classificações e tornar o processo mais didático ao usuário.

- Dashboards dinâmicos para insights em tempo real a respeito de setores específicos, ecossistemas emergentes, entre outros pontos de análise.

Visão do MVP

```python
# 5. Criar o modelo de Deep Learning

model = Sequential()
model.add(Dense(128, input_dim=len(X_train[0]), activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(64, activation='relu'))
model.add(Dense(y_train.shape[1], activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```python
# 6. Treinar o modelo

model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))
```

```
Epoch 1/10
735/735 ━━━━━━━━━━━━━━━━━━━━ 14s 17ms/step - accuracy: 0.0000e+00 - loss: 10.3175 - val_accuracy: 0.0000e+00 - val_loss: 10.5642
Epoch 2/10
735/735 ━━━━━━━━━━━━━━━━━━━━ 11s 15ms/step - accuracy: 5.8419e-05 - loss: 10.2106 - val_accuracy: 0.0000e+00 - val_loss: 11.0230
Epoch 3/10
735/735 ━━━━━━━━━━━━━━━━━━━━ 10s 14ms/step - accuracy: 4.3994e-04 - loss: 10.1255 - val_accuracy: 3.4031e-04 - val_loss: 11.4194
Epoch 4/10
735/735 ━━━━━━━━━━━━━━━━━━━━ 10s 14ms/step - accuracy: 7.3219e-04 - loss: 9.9643 - val_accuracy: 3.4031e-04 - val_loss: 11.8243
Epoch 5/10
735/735 ━━━━━━━━━━━━━━━━━━━━ 9s 12ms/step - accuracy: 0.0016 - loss: 9.7214 - val_accuracy: 3.4031e-04 - val_loss: 12.3063
Epoch 6/10
735/735 ━━━━━━━━━━━━━━━━━━━━ 8s 11ms/step - accuracy: 0.0032 - loss: 9.4535 - val_accuracy: 0.0000e+00 - val_loss: 12.7234
Epoch 7/10
735/735 ━━━━━━━━━━━━━━━━━━━━ 8s 11ms/step - accuracy: 0.0043 - loss: 9.2234 - val_accuracy: 1.7015e-04 - val_loss: 13.1961
Epoch 8/10
735/735 ━━━━━━━━━━━━━━━━━━━━ 8s 11ms/step - accuracy: 0.0075 - loss: 8.9865 - val_accuracy: 1.7015e-04 - val_loss: 13.7798
Epoch 9/10
735/735 ━━━━━━━━━━━━━━━━━━━━ 8s 11ms/step - accuracy: 0.0125 - loss: 8.7764 - val_accuracy: 8.5077e-04 - val_loss: 14.2621
Epoch 10/10
735/735 ━━━━━━━━━━━━━━━━━━━━ 8s 11ms/step - accuracy: 0.0180 - loss: 8.5493 - val_accuracy: 0.0010 - val_loss: 14.8526
```

Solução Proposta

Patent Insight Hub

- Plataforma para classificação e análise automática de patentes através do uso de *deep learning* e PLN (***BERT*/*SciBERT*** + modelos supervisionados)
- Utilização de IA (RAG + GPT-4) para justificar classificações e tornar o processo mais didático ao usuário.
- Dashboards dinâmicos para insights em tempo real a respeito de setores específicos, ecossistemas emergentes, entre outros pontos de análise.

Visão do MVP

```python
from openai import OpenAI

# Inicialize o cliente da OpenAI
client = OpenAI(api_key="sk-proj-7mSWdstuUMmMHDR3K21uxuvndhlmiQwpS_zVuz6CG_x1uN53s3NynsOfqpIoFdYrJq4lxirdOAT3BlbkFJ84C0nDcVVQqBlvoar5cT

def gerar_explicacao(abstract, subsetores_relevantes):
    subsetores_info = '\n'.join([f"Subsetor: {row['Subsetores']}, Palavras-chave: {row['KeyWords']}" for _, row in subsetores_relevantes
    prompt = f"""
    Você é um especialista em patentes. Abaixo está o resumo de uma patente e os subsetores relevantes com base nas palavras-chave.
    Resumo da patente: {abstract}
    Subsetores relevantes:
    {subsetores_info}
    Por favor, categorize essa patente no subsetor mais apropriado e forneça uma explicação detalhada sobre o motivo dessa escolha.
    """
    response = client.chat.completions.create(
        model="gpt-4-turbo-preview",  # Ou "gpt-3.5-turbo"
        messages=[
            {"role": "system", "content": "Você é um especialista em patentes."},
            {"role": "user", "content": prompt}
        ],
        max_tokens=200
    )
    return response.choices[0].message.content.strip()
# Exemplo de uso
test_abstract = df_patents['processed_abstract'].iloc[0]
subsetores_relevantes = df_subsetores
explicacao = gerar_explicacao(test_abstract, subsetores_relevantes)
print(explicacao)
```

Solução Proposta

Patent Insight Hub

- Plataforma para classificação e análise automática de patentes através do uso de *deep learning* e PLN (***BERT/SciBERT*** + modelos supervisionados)
- Utilização de IA (RAG + GPT-4) para justificar classificações e tornar o processo mais didático ao usuário.
- Dashboards dinâmicos para insights em tempo real a respeito de setores específicos, ecossistemas emergentes, entre outros pontos de análise.

Visão do MVP

```python
# 9. Dashboard com Streamlit

def streamlit_dashboard():
    st396297.title("Inteligência de Patentes com IA")
    selected_patent = st396297.selectbox("Selecione uma patente", df_patents['patent_id'])
    selected_row = df_patents[df_patents['patent_id'] == selected_patent]
    st396297.write("**Resumo:**", selected_row['processed_abstract'].values[0])
    st396297.write("**Subsetor Previsto:**", selected_row['bert_embedding'].values[0])
    st396297.write("**Explicação:**", gerar_explicacao(selected_row['processed_abstract'].values[0], df_subsetores))

if __name__ == "__main__":
    streamlit_dashboard()
```

MUITO OBRIGADO!