Relatório sobre Planejamento de Linguagem Natural (PLN)
Trabalho de Introdução à Inteligência Artificial

Universidade Estadual de Maringá - Departamento de Informática(DIN)
Introdução à Inteligência Artificial - 9788/01
Guilherme Frare Clemente - RA:124349
Marcos VInicius de Oliveira - RA:124408
Prof. Wagner Igarashi

SUMÁRIO

Introdução

Nos últimos anos, o avanço das tecnologias de processamento de linguagem natural (PLN) tem permitido que computadores compreendam e processem grandes volumes de texto de maneira cada vez mais eficiente. Uma das aplicações mais significativas dessa área é a análise de sentimentos, que consiste em identificar automaticamente as emoções, opiniões ou intenções expressas em textos. Essa técnica é amplamente utilizada por empresas para monitorar a satisfação dos clientes, entender as reações do público a produtos e serviços, e até mesmo prever tendências de mercado com base nas opiniões expressas em redes sociais e outros meios digitais.

A análise de sentimentos possui uma importância estratégica em várias indústrias. No marketing, por exemplo, empresas utilizam essa técnica para monitorar o feedback dos consumidores em tempo real, ajustando suas estratégias de acordo com as respostas emocionais dos clientes. Na área política, análises de sentimento são aplicadas para entender a percepção pública sobre candidatos ou políticas específicas, auxiliando na tomada de decisões e na elaboração de campanhas mais direcionadas. No setor de entretenimento, como é o caso das resenhas de filmes, a análise de sentimentos pode fornecer insights sobre a aceitação de obras cinematográficas, influenciando futuras produções e estratégias de distribuição.

Neste trabalho, foi desenvolvida uma aplicação de análise de sentimentos utilizando um dataset de resenhas de filmes extraído do IMDb, que é uma das maiores bases de dados online sobre filmes e séries. O objetivo principal é construir um modelo capaz de classificar as resenhas como positivas ou negativas com base no texto fornecido. Para isso, utilizou-se uma abordagem baseada em técnicas de PLN para pré-processamento dos textos, seguida de vetorização utilizando o método TF-IDF (Term Frequency-Inverse Document Frequency), e por fim, a classificação foi realizada utilizando o algoritmo de Regressão Logística. A escolha desse modelo se deu pela sua simplicidade e eficácia em tarefas de classificação binária.

A aplicação proposta neste trabalho oferece um estudo de caso relevante para entender os desafios e as possibilidades do uso de técnicas de PLN na análise de grandes volumes de texto. Além disso, o projeto explora o processo completo de desenvolvimento de um sistema de análise de sentimentos, desde a coleta e preparação dos dados até a avaliação do modelo e a análise dos resultados obtidos.

Fundamentação

O Processamento de Linguagem Natural (PLN) é um ramo da Inteligência Artificial que visa possibilitar que computadores compreendem, interpretam e respondem à linguagem humana de maneira natural. Desde o surgimento da IA, o PLN tem se consolidado como uma área de estudo crucial, dado o volume crescente de dados textuais disponíveis na internet, incluindo redes sociais, blogs, sites de notícias e bancos de dados corporativos. Essas informações textuais são frequentemente ricas em conteúdo opinativo, tornando o PLN essencial para extrair insights valiosos.

A análise de sentimentos, também conhecida como "opinion mining", é uma das aplicações mais proeminentes do PLN. Ela envolve a categorização de opiniões expressas em um texto em categorias predefinidas, como positivas, negativas ou neutras. O processo de análise de sentimentos geralmente começa com o pré-processamento dos dados, que pode incluir a remoção de ruídos (como HTML tags e caracteres especiais), normalização do texto (por exemplo, converter para minúsculas), remoção de stopwords (palavras comuns que não contribuem significativamente para o significado do texto), e a lematização, que transforma palavras em sua forma base.

Após o pré-processamento, o próximo passo é a vetorização do texto, que é a transformação do texto em uma representação numérica que possa ser utilizada por algoritmos de machine learning. Uma das técnicas mais usadas para essa finalidade é o TF-IDF (Term Frequency-Inverse Document Frequency). O TF-IDF avalia a importância de uma palavra em um documento dentro de um corpus, balanceando a frequência da palavra no documento com sua frequência no corpus inteiro. Isso resulta em vetores que representam cada documento, prontos para serem usados por modelos de aprendizado de máquina.

Na sequência, um modelo de aprendizado de máquina é treinado para classificar os textos vetorizados. Dentre os vários algoritmos disponíveis, a Regressão Logística é frequentemente escolhida para tarefas de classificação binária devido à sua simplicidade e eficácia. A Regressão Logística modela a probabilidade de uma determinada classe como uma função linear das características de entrada, permitindo que o modelo faça previsões sobre a classe de sentimento de novos textos com base nos padrões aprendidos durante o treinamento.

O uso de modelos de machine learning em análise de sentimentos, no entanto, não está isento de desafios. Por exemplo, a detecção de sarcasmo, ironia ou sentimentos ambíguos pode ser particularmente difícil, mesmo para modelos avançados. Além disso, a qualidade e a representatividade dos dados de treinamento são fatores críticos que podem influenciar significativamente o desempenho do modelo.

Neste trabalho, a análise de sentimentos foi aplicada ao dataset de resenhas de filmes do IMDb, utilizando o modelo de Regressão Logística. Este estudo oferece uma visão sobre como técnicas clássicas de PLN e machine learning podem ser combinadas para resolver problemas reais de classificação de texto. A metodologia adotada demonstra a importância de um pré-processamento cuidadoso e a seleção apropriada de características para a construção de modelos eficazes.

Metodologia

A metodologia deste trabalho foi cuidadosamente planejada para garantir uma abordagem estruturada e eficiente na análise de sentimentos em textos. O processo foi dividido em quatro etapas principais: coleta de dados, pré-processamento, treinamento do modelo e avaliação.

1. **Coleta de Dados:** O dataset utilizado foi obtido do Kaggle, uma plataforma de datasets públicos amplamente reconhecida. O dataset selecionado foi o "IMDb Dataset of 50K Movie Reviews", que contém 50.000 resenhas de filmes, categorizadas como "positivas" ou "negativas". Essa base de dados foi escolhida devido à sua popularidade e ao fato de fornecer uma distribuição balanceada de classes, essencial para o treinamento adequado do modelo de machine learning.

2. **Pré-processamento:** O pré-processamento dos dados textuais é uma etapa crítica em qualquer tarefa de Processamento de Linguagem Natural. No presente trabalho, essa etapa incluiu a remoção de ruídos no texto, como tags HTML e caracteres especiais, a conversão das palavras para minúsculas, a remoção de palavras irrelevantes (stopwords) e a lematização, que converte as palavras para sua forma base. Em seguida, os textos foram transformados em vetores numéricos utilizando a técnica TF-IDF, que avalia a importância de cada palavra no contexto de todo o corpus.

3. **Treinamento do Modelo:** Após o pré-processamento, os dados foram divididos em conjuntos de treino e teste, utilizando uma proporção de 80% para o treino e 20% para o teste, com estratificação para manter a proporção das classes. O modelo de Regressão Logística foi então treinado utilizando os dados vetorizados. A escolha da Regressão Logística se deve à sua simplicidade e eficácia em problemas de classificação binária, como o que é abordado neste trabalho.

4. **Avaliação:** A avaliação do modelo foi realizada através da aplicação do conjunto de teste, utilizando métricas padrão como acurácia, matriz de confusão e relatório de classificação (precision, recall, F1-score). Além disso, foram realizados testes com frases de exemplo para verificar a eficácia do modelo em situações práticas e avaliar sua performance em casos específicos.

Desenvolvimento

O desenvolvimento do projeto foi realizado no ambiente Google Colab, utilizando a linguagem Python, conhecida por sua vasta gama de bibliotecas especializadas em machine learning e processamento de linguagem natural.

1. **Coleta e Preparação dos Dados:** A primeira etapa envolveu o download e carregamento do dataset de resenhas de filmes do IMDb, diretamente do Kaggle. Após a análise exploratória inicial, foi verificado que o dataset estava balanceado, com uma distribuição quase igual de resenhas positivas e negativas.

2. **Pré-processamento dos Textos:** Utilizou-se uma função de limpeza de texto desenvolvida especificamente para este projeto. Esta função remove tags HTML, caracteres especiais e stopwords, e aplicou a lematização das palavras. Após a limpeza, os textos foram transformados em vetores numéricos utilizando TF-IDF, limitando-se às 5000 palavras mais relevantes, a fim de reduzir a dimensionalidade e focar nas características mais significativas.

3. **Treinamento e Avaliação do Modelo:** O modelo de Regressão Logística foi treinado com os dados vetorizados e, posteriormente, testado com o conjunto de testes. A acurácia do modelo foi satisfatória, com resultados positivos também em termos de precision, recall e F1-score. A matriz de confusão mostrou que o modelo conseguiu classificar corretamente a maioria dos exemplos, com poucos erros de classificação.

4. **Testes com Frases de Exemplo:** Para validar o modelo em situações reais, foram selecionadas algumas frases representativas de diferentes sentimentos. Essas frases foram pré-processadas e classificadas pelo modelo. A maioria das previsões foi coerente com o sentimento esperado, demonstrando a capacidade do modelo de generalizar para novos exemplos.

Análise Crítica do algoritmo PLN

Pontos Fortes:

- **Simplicidade e Eficiência:** O modelo utilizado é um classificador de Regressão Logística, que é uma abordagem simples e eficiente para tarefas de análise de sentimento. A simplicidade do modelo permite uma rápida execução e interpretação dos resultados. No código, isso é evidenciado pela criação do modelo com **LogisticRegression()** e o treinamento com **model.fit(X_train_vec, y_train).**

```
# Passo 6: Treinamento do modelo de Regressão Logística
model = LogisticRegression()
model.fit(X_train_vec, y_train)

# Passo 7: Avaliação do modelo
y_pred = model.predict(X_test_vec)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```

- **Resultados Satisfatórios:** O modelo demonstrou resultados satisfatórios para a maioria das frases de teste, como evidenciado pelos resultados impressos com **print(classification_report(y_test, y_pred))** e **print(confusion_matrix(y_test, y_pred)).** Os valores de precisão, recall e f1-score para as classes positiva e negativa são razoavelmente altos.

```
# Passo 7: Avaliação do modelo
y_pred = model.predict(X_test_vec)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```

Pontos Fracos:

- **Captação de Nuances Complexas:** Embora o modelo seja eficiente, ele pode não captar nuances mais complexas do texto, como sarcasmo ou ironia. Isso é uma limitação da abordagem de Regressão Logística e do pré-processamento básico utilizado. A função de limpeza do texto (**clean_text**) remove todas as pontuações e palavras de parada, o que pode simplificar demais o contexto das frases.

```
# Função de limpeza do texto
def clean_text(text):
    text = re.sub(r'<[^>]+>', ' ', text)
    text = re.sub(r'[^a-zA-Z]', ' ', text)
    text = text.lower()
    text = text.split()
    text = [word for word in text if not word in stopwords.words('english')]
    text = ' '.join(text)
    return text
```

- **Dependência da Qualidade dos Dados:** A performance do modelo depende fortemente da qualidade e da representatividade dos dados de treinamento. Se os dados contiverem vieses ou erros, isso pode se refletir na precisão do modelo. A análise dos dados foi realizada com **print(df['sentiment'].value_counts())**, o que ajuda a entender a

distribuição, mas não garante que todos os aspectos do sentimento sejam capturados adequadamente.

```
# Passo 2: Carregar e analisar os dados
df = pd.read_csv("IMDB Dataset.csv")
print(df.head())
print(df['sentiment'].value_counts())
```

Oportunidades de Melhorias:

- **Modelos mais Complexos:** Para melhorar a capacidade do modelo de entender contextos mais complexos, é possível utilizar modelos mais avançados como BERT (Bidirectional Encoder Representations from Transformers) ou LSTM (Long Short-Term Memory). Estes modelos são mais eficazes na captura de nuances semânticas e sintáticas do texto.
- **Aumento do DataSet:** Aumentar o tamanho e a diversidade do dataset pode melhorar o desempenho do modelo. A inclusão de mais exemplos de frases e a consideração de diferentes estilos de escrita e jargões pode ajudar o modelo a generalizar melhor para novos dados.

Abrangência e Aplicação:

- **Aplicabilidade:** O modelo é aplicável em diversas áreas como sistemas de recomendação, análise de comentários e monitoramento de redes sociais. Ele pode ser usado para automatizar a análise de sentimentos em grande escala e fornecer insights sobre as opiniões dos usuários.
- **Limitações:** O modelo pode ter dificuldades em lidar com textos irônicos e sarcásticos, como demonstrado por frases que não corresponderam às expectativas. Frases como "I expected a better plot twist, but it was fine overall," onde o sentimento esperado era neutro, foram classificadas erroneamente como positivas. Isso destaca a necessidade de modelos mais sofisticados para lidar com esses casos.

Exemplos de Frases onde o Modelo Falhou:

- **Frase: “I expected a better plot twist, but it was fine overall.”**
  - Sentimento Esperado: Neutro
  - Sentimento Predito: Positivo

Este exemplo foi identificado na seção de análise de sentimento do código, onde **sentiment_predictions** foi usado para prever o sentimento das frases de teste. A frase em questão foi mencionada para ilustrar uma falha na capacidade do modelo de captar o sentimento neutro corretamente.

```
# Passo 8: Testar com frases de exemplo
sentences = [
    "I love this movie, it was fantastic!",
    "The film was horrible, I hated it.",
    "An excellent performance by the lead actor.",
    "The plot was boring and predictable.",
    "I enjoyed the cinematography and the music was great."
]

cleaned_sentences = [clean_text(sentence) for sentence in sentences]
vectorized_sentences = vectorizer.transform(cleaned_sentences)
sentiment_predictions = model.predict(vectorized_sentences)

for sentence, sentiment in zip(sentences, sentiment_predictions):
    print(f"Sentence: {sentence}\nPredicted Sentiment: {sentiment}\n")

# Imprimir análise crítica
print_analysis()
```

Conclusões

O trabalho apresentou uma aplicação prática de análise de sentimentos utilizando técnicas de Processamento de Linguagem Natural e Machine Learning. O modelo de Regressão Logística, treinado com um dataset de resenhas de filmes do IMDb, demonstrou boa capacidade de classificação, com resultados consistentes nas métricas de avaliação. A análise de frases de exemplo também confirmou a eficácia do modelo em situações práticas.

No entanto, algumas limitações foram observadas, como a dificuldade do modelo em lidar com frases contendo sarcasmo ou ambiguidade, que são desafios conhecidos na análise de sentimentos. Futuras melhorias poderiam incluir a experimentação com modelos mais complexos, como redes neurais ou modelos de linguagem pré-treinados, como BERT, que podem capturar melhor as nuances do idioma.

Em resumo, este trabalho destacou a importância de um pré-processamento cuidadoso e a escolha adequada de técnicas de vetorização e modelagem para alcançar bons resultados em tarefas de análise de sentimentos. O modelo desenvolvido pode ser aplicado em diversos contextos onde a compreensão automática das opiniões expressas em textos é relevante.

Referências Bibliográficas

Kaggle. "IMDb Dataset of 50K Movie Reviews." Disponível em: https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011). "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research, 12, 2825–2830.

Manning, C., Raghavan, P., Schütze, H. (2008). "Introduction to Information Retrieval." Cambridge University Press.