



Universidade do Minho

Universidade do Minho

**Mestrado em Engenharia Informática
Aprendizagem Profunda (2022/2023)**

**PROCESSAMENTO DE TEXTOS COM
FEW-SHOT, ONE-SHOT E ZERO-SHOT
LEARNING APLICADO A LETRAS DE
MÚSICAS**

Grupo N.º2

Anabela Pereira PG49995

Cláudia Ribeiro PG49998

Sara Marques PG47657

Github: https://github.com/alpereirinha/DL_Project

Braga, 12 de junho de 2023

Conteúdo

1	Introdução	2
2	<i>Dataset</i>	3
3	Modelos Aplicados	4
3.1	GPT	4
3.1.1	Modelo Utilizado	4
3.1.2	<i>Zero-shot Learning</i>	4
3.1.3	<i>One-shot Learning</i>	4
3.1.4	<i>Few-shot Learning</i>	5
3.2	Rede Neuronal	5
3.2.1	Implementação	5
3.2.2	Resultados	6
3.3	NLTK	7
3.3.1	Implementação	7
3.3.2	Resultados	7
4	Análise dos Resultados	8
5	Conclusão	9

1. Introdução

De modo a explorar em termos práticos quais as potencialidades e utilidades das técnicas de *Deep Learning* abordadas ao longo deste semestre, desenvolveu-se o seguinte trabalho a fim de comparar diferentes modelos partindo do conhecimento presente num *dataset*, de modo a retirar conclusões sobre a sua eficiência e nível de exatidão.

Para este projeto foi utilizado um *dataset* de letras de músicas, sendo o nosso objetivo identificar o sentimento associado a cada letra através da aplicação de diferentes modelos e utilizando as técnicas abordadas no trabalho de investigação anterior: *few-shot*, *one-shot* e *zero-shot learning*.

Serão então analisados neste relatório os resultados obtidos através da aplicação destes métodos num modelo GPT, e comparados com os resultados obtidos através de uma rede neuronal e de um modelo pré-treinado do NLTK.

2. *Dataset*

Originalmente o *dataset* utilizado para o projeto continha os dados relativos ao top 100 de músicas de cada ano entre 2012 e 2022. Visto que estes dados não incluíam os valores de sentimento associados a cada música, foi aplicado um modelo de NLTK de modo a calculá-los. No entanto, devido a preocupações relativas ao nível de exatidão providenciado pelo NLTK, que poderia não fornecer os melhores resultados para uso como base comparativa, procurou-se antes utilizar um *dataset* que já incluísse o sentimento de cada música, ou seja, *pre-labeled*, e reter a abordagem com NLTK tratando este apenas como um outro modelo a ser comparado com os restantes.

O *dataset* que acabou por ser utilizado foi, portanto, obtido através da plataforma Kaggle, sendo composto por 150 mil músicas, todas elas em inglês, e contendo para cada uma a sua letra, artista, título e valor de sentimento associado, obtido através da API Valence do Spotify. Sendo o nosso objetivo identificar o sentimento associado a cada letra, estes serão os dois parâmetros mais relevantes a ter em conta.

Os valores dos sentimentos no *dataset* variam entre 0 e 1, sendo que consideramos valores inferiores a 0.4 como correspondentes a um sentimento negativo, superiores a 0.6 correspondentes a um sentimento positivo, e os restantes como apresentando um sentimento neutro. Tendo isto em conta, podemos observar que o *dataset* apresenta a seguinte distribuição de músicas positivas, negativas e neutras.

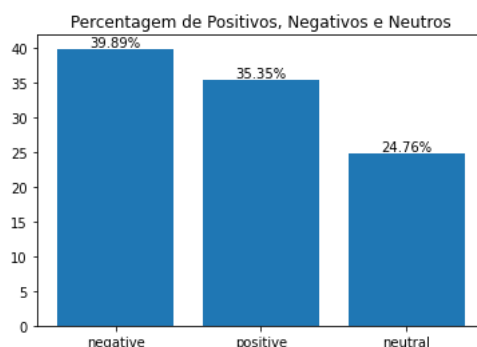


Figura 2.1: Distribuição de sentimentos no *dataset*

No entanto, deve ser apontado que estes valores de sentimento, tendo sido adquiridos pela API Valence, levam em consideração o áudio da música e não apenas a sua letra. Como tal, sendo o objetivo deste projeto fazer previsões baseadas exclusivamente no conteúdo das letras das músicas, poderá haver discrepâncias que irão influenciar negativamente o nível de exatidão dos resultados obtidos.

3. Modelos Aplicados

3.1 GPT

3.1.1 Modelo Utilizado

O principal modelo a ser utilizado, com vista a aplicar as abordagens de *few-shot*, *one-shot* e *zero-shot learning*, será o GPT, mais especificamente, o *text-davinci-003*.

Apesar do modelo GPT mais recente, GPT-4, proporcionar mais funcionalidades na área de processamento de linguagem natural e um maior nível de exatidão, este não se encontra atualmente disponível para uso gratuito, tendo por isso levado a que o grupo optasse pela utilização de uma versão anterior - neste caso, utilizou-se o modelo *text-davinci-003* para as três abordagens. Isto, pois apesar do modelo GPT-3.5 ser o mais indicado para casos de *zero-shot learning*, o modelo *davinci* é mais adequado para abordagens de *k-shot learning* (ou seja, *one-shot* e *few-shot learning*), e é também menos custoso nas chamadas à API.

Devido às limitações no uso da API do GPT de modo gratuito, serão também introduzidas apenas 100 músicas do *dataset* original ao utilizar este modelo.

3.1.2 *Zero-shot Learning*

De modo a aplicar a abordagem de *zero-shot learning*, foi fornecida ao modelo GPT a seguinte *prompt*:

"Analyze the following lyrics and determine if their sentiment is either 'positive', 'negative' or 'neutral'. Format your answer as a CSV file with a column for the sentiment category and another for the lyrics. Lyrics to analyze: (...)"

Neste caso, não foi, portanto, fornecido qualquer exemplo para cada uma das categorias "positive", "negative" ou "neutral" na fase de treino. O modelo irá então classificar cada *input* partindo apenas da informação textual providenciada pela *prompt*, que será interpretada pelo GPT.

Com esta abordagem, foi obtida uma exatidão dos resultados de 36%, tendo ocorrido 36 acertos. Este baixo valor poderá ser atribuído à falta de informação introduzida na *prompt* que possa servir de orientação ao modelo sobre como melhor classificar cada *input*.

3.1.3 *One-shot Learning*

De modo a aplicar a abordagem de *one-shot learning*, foi fornecida ao modelo GPT a seguinte *prompt*:

*“The following lyrics is POSITIVE: [example-1].
The following lyrics is NEGATIVE: [example-2].
The following lyrics is NEUTRAL: [example-3].
Analyze the following lyrics and determine if their sentiment is either POSITIVE,
NEGATIVE or NEUTRAL: (...)”*

Neste caso, foi fornecido um único exemplo representativo de cada categoria na fase de treino, de acordo com a abordagem *one-shot*. Estes exemplos foram retirados do conjunto de dados de treino (*dataset_train.csv*) de modo a que não reaparecessem no conjunto de dados de teste.

Com esta abordagem foi obtida uma exatidão dos resultados de 39%, tendo ocorrido 39 acertos. Podemos então observar que o modelo beneficiou da introdução de exemplos a partir dos quais basear a sua classificação de dados, considerando o ligeiro aumento do nível de exatidão em comparação à abordagem de *zero-shot learning*.

3.1.4 *Few-shot Learning*

De modo a aplicar a abordagem de *few-shot learning*, foi fornecida ao modelo GPT a seguinte *prompt*:

*“The following lyrics is POSITIVE: [example-1].
The following lyrics is NEGATIVE: [example-2].
The following lyrics is NEUTRAL: [example-3].
The following lyrics is POSITIVE: [example-4].
The following lyrics is NEGATIVE: [example-5].
The following lyrics is NEUTRAL: [example-6].
Analyze the following lyrics and determine if their sentiment is either POSITIVE,
NEGATIVE or NEUTRAL: (...)”*

Neste caso, foi fornecido mais que um exemplo para cada categoria na fase de treino, permitindo que o modelo identifique semelhanças entre exemplos dados para uma mesma categoria e diferenças com os exemplos restantes, obtendo assim um conhecimento mais amplo e específico do que constitui cada categoria. À semelhança da abordagem de *one-shot learning*, os casos de exemplo foram retirados de *dataset_train.csv*.

Esta abordagem obteve os melhores resultados entre as três, apresentando um nível de exatidão de 46%, tendo ocorrido 46 acertos.

3.2 Rede Neuronal

3.2.1 Implementação

A rede neuronal desenvolvida é uma rede neuronal recorrente (RNN), onde foram aplicadas as seguintes camadas:

- **Embedding** - Responsável por vetorizar os dados de entrada;
- **Dropout** - Aplicado de modo a reduzir o *overfitting* do modelo, com um valor de 0,7;
- **LSTM** - Arquitetura adequada para o processamento de informação sequencial, como é o caso de texto, em que cada palavra irá influenciar o significado (e, neste caso, sentimento) das palavras seguintes. Foi aplicada regularização L2 nesta camada de modo a reduzir o *overfitting* do modelo;

- **Softmax** - Função aplicada na camada de ativação de modo a preservar os valores necessários para identificar as três categorias de sentimento: positivo, negativo e neutro.

Para além disto, de modo a melhorar a exatidão dos resultados, as entradas foram lematizadas (ou seja, foi ignorada a flexão das palavras nelas contidas de modo a agrupar as que têm o mesmo significado) e filtradas de modo a remover palavras sem significado antes do treino e teste do modelo.

	lyrics	sentiment
0	No, no\ \n i ain't ever trapped out the bando\ r...	positive
1	The drinks go down and smoke goes up, I feel m...	positive
2	She don't live on planet Earth no more\ \n She ...	negative
3	Trippin' off that Grigio, mobbin', lights low\ ...	neutral
4	I see a midnight panther, so gallant and so br...	negative

Figura 3.1: Entradas originais

	lyrics	sentiment
0	no no i aint ever trapped out the bando but ...	positive
1	the drinks go down and smoke goes up i feel my...	positive
2	she dont live on planet earth no more she fou...	negative
3	trippin off that grigio mobbin lights low tri...	neutral
4	i see a midnight panther so gallant and so bra...	negative

Figura 3.2: Entradas após tratamento dos dados

O *dataset* usado foi também balanceado de modo a ter a mesma proporção de músicas negativas, positivas e neutras.

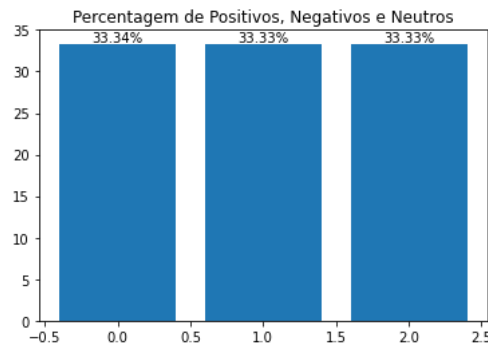


Figura 3.3: Distribuição de sentimentos no *dataset* após balanceamento

Relativamente aos dados utilizados, 80% do *dataset* foi utilizado para treino, e 20% para teste. Destes dados de teste, 10% foram utilizados para validação dos resultados.

3.2.2 Resultados

Com estes parâmetros, foi obtido um nível de exatidão de 46%. Este valor foi obtido após ajustes aos valores de *dropout*, *learning rate* e regularização de modo a diminuir o *overfitting* do modelo. Foi também testado um modelo com todas as mesmas características mas aplicando SimpleRNN em lugar de LSTM, não se tendo verificado uma melhoria dos resultados.

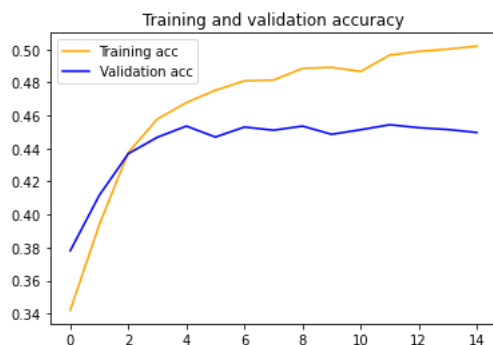


Figura 3.4: Progressão da *accuracy*

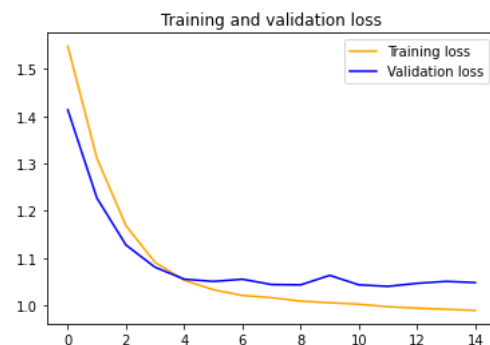


Figura 3.5: Progressão da *loss*

3.3 NLTK

3.3.1 Implementação

O plataforma de Python NLTK inclui *libraries* de processamento de texto que poderão ser utilizadas para identificar o sentimento associado a letras de músicas, tais como tem sido efetuado com os modelos anteriores. Como tal, iremos utilizar o modelo de análise de sentimento Vader do NLTK e analisar os resultados obtidos.

Ao ser aplicado ao *dataset*, o Vader irá identificar o valor de sentimento de cada palavra de cada letra, e o sentimento da letra em si será calculado a partir do valor das várias palavras que a compõem. No entanto, isto exige que, em primeiro lugar, sejam removidas das letras todas as palavras a partir das quais o Vader não consegue extrair significado, de modo a obter resultados mais exatos. Como tal, serão removidas das letras palavras e onomatopeias como "oh", "na", "yeah", etc, antes de o Vader lhes ser aplicado.

Após a realização desta filtragem foi aplicado o modelo Vader do NLTK. Tendo em conta que este irá apresentar valores de sentimento entre -1 e 1, serão considerados valores inferiores a -0,2 como negativos, superiores a 0,2 como positivos, e os restantes como neutros, identificando assim cada tipo de sentimento de acordo com as mesmas proporções que o *dataset* original.

3.3.2 Resultados

Com esta abordagem, foi obtido um nível de exatidão de 37%. Sendo o modelo Vader pré-treinado, é possível que este valor de exatidão seja baixo devido a limitações no treino do próprio modelo.

Para além disto, o Vader não leva em consideração a sequência das palavras contidas em cada música, o que poderá resultar em valores de sentimento atribuídos de forma errada devido à falta de contextualização das palavras (por exemplo, "*sad*" terá sempre um valor de sentimento negativo, mesmo quando inserido numa frase que nega o adjetivo, como "*not sad*").

4. Análise dos Resultados

Após a aplicação dos vários modelos, podemos observar os diferentes resultados obtidos na seguinte tabela.

GPT			RNN	NLTK Vader
<i>zero-shot</i>	<i>one-shot</i>	<i>few-shot</i>		
36%	39%	46%	46%	37%

A correlação entre os diversos níveis de exatidão obtidos e as abordagens tomadas corresponde às expectativas, sendo que as abordagens de *zero-shot learning* ou com recurso ao NLTK Vader apresentam os valores mais baixos comparativamente aos restantes, e o *few-shot learning* e a RNN os mais altos.

No entanto, todos estes valores são relativamente baixos. Esta falha é devida ao *dataset* utilizado, cujos valores de sentimento foram atribuídos tendo em conta não só o conteúdo das letras mas também o áudio das músicas. Como tal, é possível que uma letra negativa, sobreposta com uma melodia muito positiva, leve a que a música em questão seja classificada como positiva, o que nunca seria identificado corretamente pelos nossos modelos visto estes considerarem apenas o conteúdo textual das letras.

5. Conclusão

Consideramos o trabalho desenvolvido satisfatório, tendo este servido como um método eficaz de consolidar as temáticas abordadas na UC de Aprendizagem Profunda ao longo do semestre, para além das abordagens de *few-shot*, *one-shot* e *zero-shot learning* que foram analisadas no trabalho de investigação anterior, nomeadamente no contexto do processamento de linguagem natural.

Como trabalho futuro, seria relevante a repetição destas abordagens com um outro *dataset*, ou a exploração de outros modelos como o BERT, de modo a produzir uma maior variedade de dados comparativos. Seria também possível explorar as aplicações de *transfer learning* no contexto de uma rede neuronal, podendo este ser aplicado de modo a obter um outro exemplo de uma abordagem de *zero-shot learning*.

Em geral, consideramos o balanço do trabalho desenvolvido como positivo, apesar do seu potencial para expansão e ajustes, tendo este sofrido de uma escolha imprópria de *dataset*.

Bibliografia

- [1] Documentação Spotify API: <https://developer.spotify.com/documentation/web-api/reference/get-audio-features>
- [2] Documentação GPT: <https://platform.openai.com/docs/guides/gpt>
- [3] Documentação NLTK Vader: https://www.nltk.org/_modules/nltk/sentiment/vader.html