

INSTITUTO FEDERAL
Goiás

Projeto Prático PLN - Processamento de Linguagem Natural

Prof. Dr. Daniel Xavier Souza

Prof. Dr. Sérgio Daniel Canuto

Alunos Grupo 4:

Cleibson de Oliveira

Wagner da Silva

Marcos Rodrigues Brugnaro

1. Adaptações

- Definições das bases de dados e transformers definidas no documento do Projeto:

	Grupo 4
Base pequena	Buscapé
Base média	B2W
TLM Pequeno	Ro_B
TLM Grande	Ro_L
TLM multilíngua	Bert-L

- Adaptações para o desenvolvimento do projeto:

	Grupo 4
Base Média	Buscapé
Base Pequena	B2W
TLM Pequeno	Ro_B
TLM Grande	Ro_L
TLM Multilíngue	Bert_B

2. Pré-Processamento das bases Buscapé e B2W

2. Base Buscapé - Pré-Processamento

❖ **Verificação de quantidades de instâncias e classes (features)**, do dataset original.

- Instâncias: 84.991;
- Classes (features): 5;

❖ **Definição da Tarefa:** Classificação de ratings das avaliações de clientes:

- 4 e 5: Positiva;
- 1 e 2: Negativa;
- 3: Descartadas;

❖ Tendo em vista estas regras, **avaliações positivas recebem valor 1 e negativas recebem valor 0;**

❖ **Remoção de instâncias nulas:**

- Após a importação do dataset, identificamos uma instância nula na feature `review_text`, removemos a mesma;

	<code>review_text</code>	<code>rating</code>
38852	NaN	4

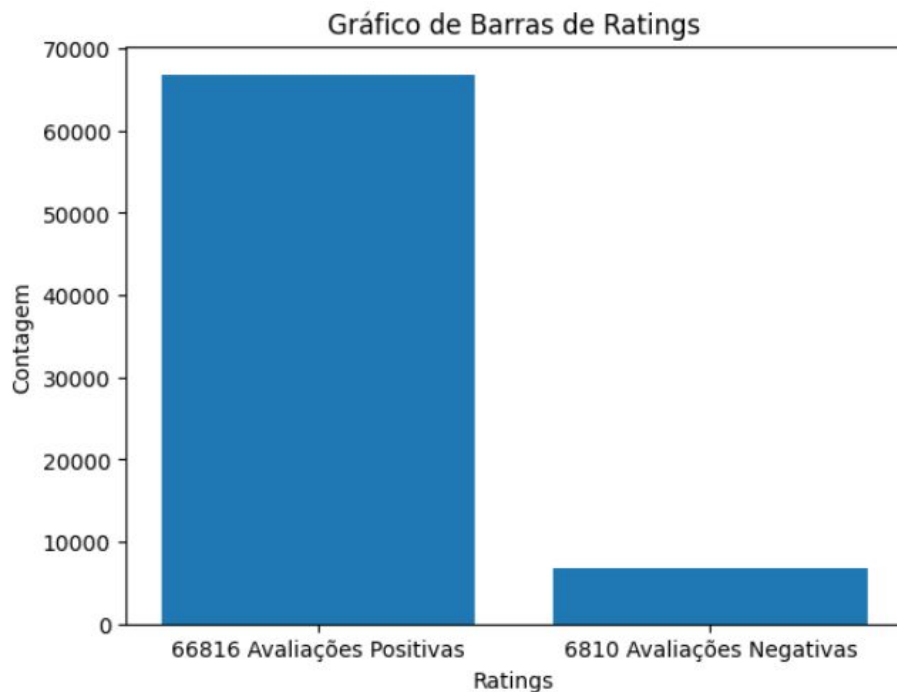
2. Base Buscapé - Pré-Processamento

- ❖ Padronização do texto em lowercase (letras minúsculas);
- ❖ Remoção de caracteres especiais;
- ❖ Substituição de ç por c;
- ❖ Remoção de hiperlinks;
- ❖ Descarte de instâncias com avaliação igual a 3;
- ❖ Ajustando ratings (binarização):
 - ratings 1 e 2 = 0
 - ratings 4 e 5 = 1

2. Base Buscapé - Pré-Processamento

❖ Balanceamento das classes:

- Identificamos que as classes estavam desbalanceadas, 66.816 avaliações positivas e 6.810 negativas;

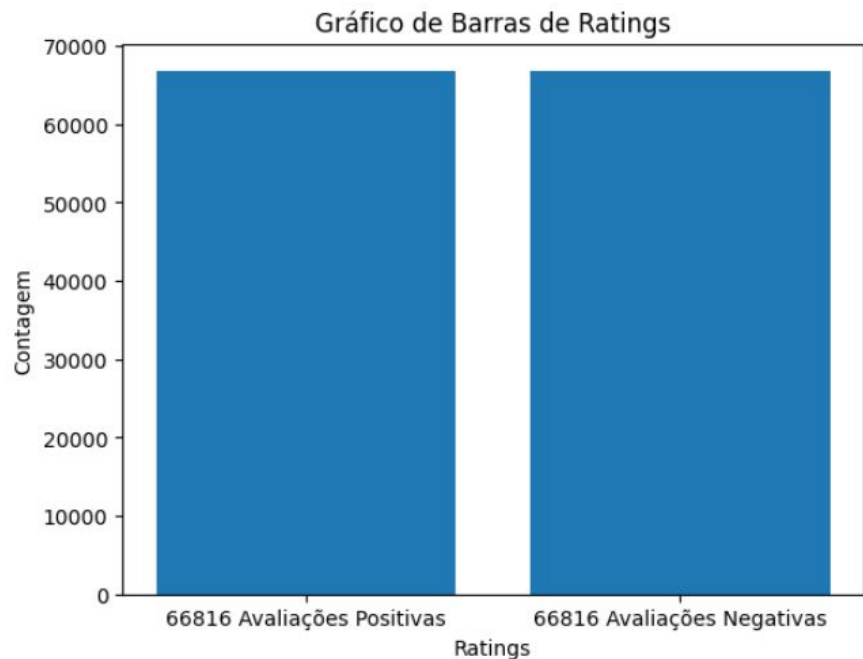


- Para resolver este problema utilizamos a técnica de reamostragem de dados RandomOverSampler;
- Geração de novas amostras aleatórias para classe minoritária;
- Desta forma a técnica tenta igualar o número de amostras da classe minoritária ao número de amostras da majoritária;

2. Base Buscapé - Pré-Processamento

❖ Balanceamento das classes:

- Resultado após o balanceamento das classes:



- Após o balanceamento das classes a base buscapé:
 - Ocorreu aumento da quantidade de instâncias para 133.632;
 - 66.816 avaliações positivas e 66.816 avaliações negativas;
- Após este processo salvamos os dados pré-processados em um dataframe que iremos utilizar durante os treinamentos;

2. Base B2W - Pré-Processamento

- ❖ Verificação de quantidades de instâncias e classes (features), do dataset original.
 - Instâncias: 132.373;
 - Classes (features): 5;
- ❖ Definição da Tarefa: Classificação de ratings das avaliações de clientes:
 - 4 e 5: Positiva;
 - 1 e 2: Negativa;
 - 3: Descartadas;
 - Tendo em vista estas regras, avaliações positivas recebem valor 1 e negativas recebem valor 0;
- ❖ Remoção de instâncias nulas:
 - Não ocorreram instâncias nulas;
- ❖ Padronização do texto em lowercase (letras minúsculas);

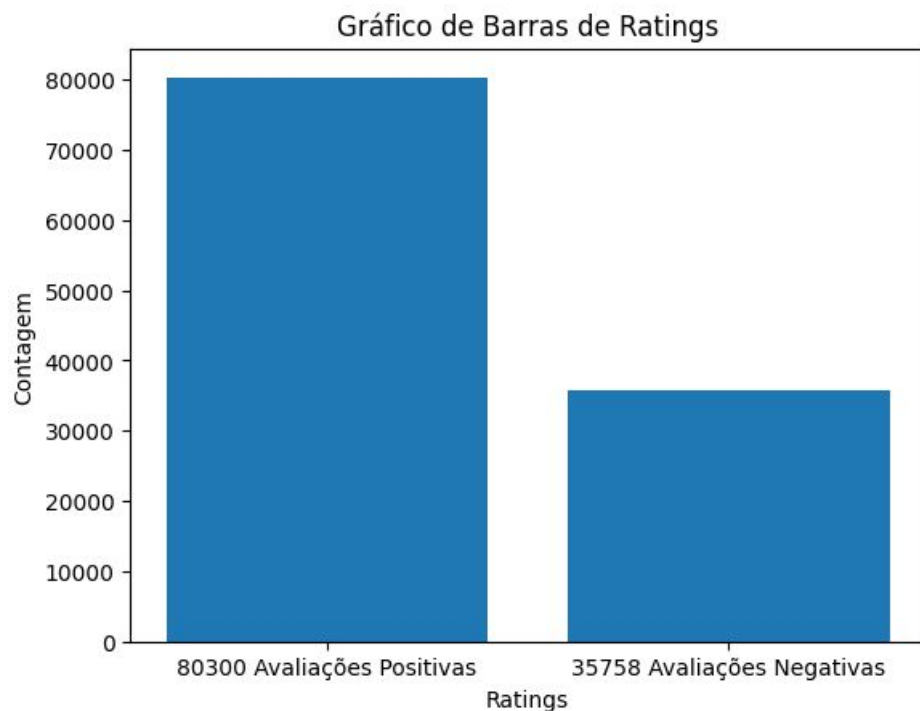
2. Base B2W - Pré-Processamento

- ❖ Remoção de caracteres especiais e substituição de ç por c;
- ❖ Remoção de hiperlinks;
- ❖ Descartando instâncias com avaliação igual a 3;
- ❖ Ajustando ratings (binarização):
 - 1 e 2 = 0
 - 4 e 5 = 1

2. Base B2W - Pré Processamento

❖ Balanceamento das classes:

- Identificamos que as classes do dataset B2W estavam desbalanceadas, 80.300 avaliações positivas e 35.758 negativas;

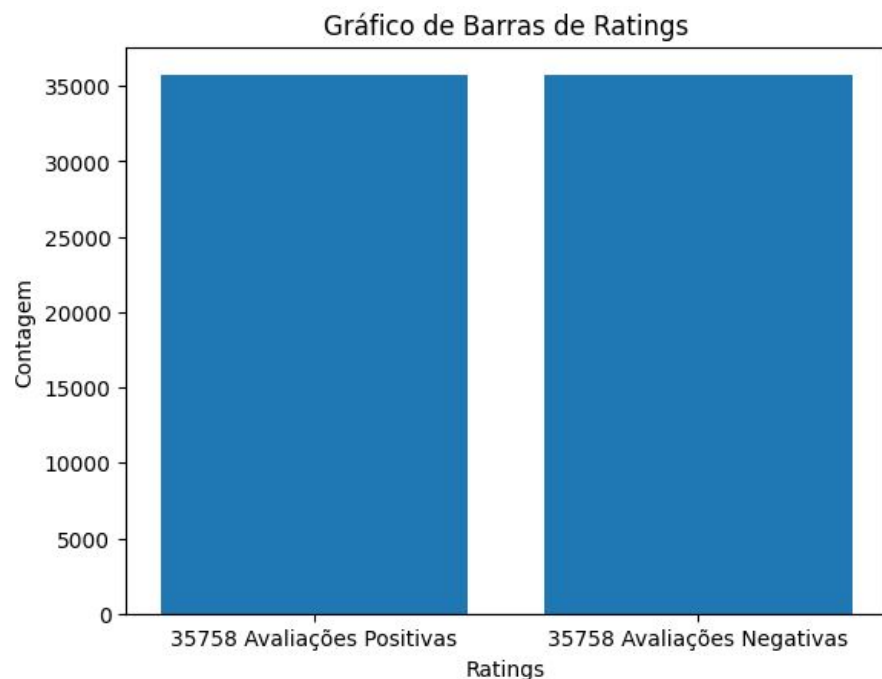


- Para resolver este problema utilizamos a técnica de reamostragem de dados RandomUnderSampler;
- Desta forma a técnica tenta igualar o número de amostras da classe majoritária ao número de amostras da classe minoritária;

2. Base B2W - Pré-Processamento

❖ Balanceamento das classes:

- Resultado após o balanceamento das classes:



- Após o balanceamento das classes a base B2W:
 - Ocorreu a diminuição da quantidade de instâncias para 71.516;
 - Sendo 37.758 avaliações positivas e 37.758 avaliações negativas;
- Após este processo salvamos os dados pré-processados em um dataframe que iremos utilizar durante os treinamentos;

3. Bases de Dados

- ❖ Consolidado de informações originais e após pré-processamento, binarização e balanceamento das bases:

Detalhamento das Bases de Dados			
Dataset Original			
Dataset	Quantidade de Instâncias (Dataset Original)	Classes	Quantidade de Instâncias Nulas
Buscapé	84.991	5	1
B2w	132.373	5	0

Detalhamento das Bases de Dados					
Dataset Pré Processado					
Dataset	Binarização (Quantidade de Classes Positivas)	Binarização (Quantidade de Classes Negativas)	Técnica de balanceamento	Quantidade de Instâncias após Binariazação, Pré-processamento e Balanceamento	Quantidade de Classes após Binariazação, Pré-processamento e Balanceamento
Buscapé	66.816	66.816	RandomOverSampler	133.632	2
B2w	35.758	35.758	RandomUnderSampler	71.516	2

4. Detalhamento da execução dos Experimentos - Embeddings Estáticos

◆ Definição dos modelos:

- ❖ TFIDF + MLP
- ❖ GloVe + MLP
- ❖ GloVe + Fine Tuning
- ❖ FastText + MLP
- ❖ FastText + FineTuning

◆ Variação dos seguintes hiperparâmetros:

- ❖ Variação de Dropout: 5%, 15%, 25%, 35%, 50%
- ❖ Variação de Learning Rate: 5e-4, 1e-3, 5e-3, 1e-2

◆ Hiperparâmetros que mantemos durante os treinamentos:

- ❖ Cross-validation, 10 k-folds;
- ❖ Épocas: 5;

◆ Arquitetura rede MLP:

- ❖ Camada densa 24 neurônios, função relu;
- ❖ Dropout com respectiva variação;
- ❖ Camada densa com 1 neurônio, função de ativação Sigmoid;
- ❖ Sumarizamos a rede para identificarmos total de parâmetros treináveis do modelo;

5. Treinamento dos modelos - Embeddings Estáticos

Exemplo de sequência de variações de hiperparâmetros para cada modelo de Embeddings Estáticos:

❖ **Variação de hiperparâmetros:**

Dropout 5%, Learning Rate: 5e-4

Dropout 5%, Learning Rate: 1e-3

Dropout 5%, Learning Rate: 5e-3

Dropout 5%, Learning Rate: 1e-2

❖ **Iniciando outra sequência de variações:**

Dropout 15%, Learning Rate: 5e-4

❖ **E segue a mesma lógica até finalizar as variações:**

- 5 sequências de variações (5 variações de dropout)
- Cada variação com 4 treinamentos (4 variações de learning rate)
- Para cada um dos 5 modelos;
- 100 treinamentos experimentais para cada uma das bases;
- 200 treinamentos abrangendo as duas bases;

5. Treinamento dos modelos - Embeddings Estáticos

Observações em treinamentos dos modelos de embeddings estáticos:

- Pré-processamento dividido em duas etapas:
- A primeira etapa foi descrita anteriormente até a finalização do balanceamento das classes;
- A segunda etapa descrita em seguida:

Segunda etapa - particularidades dos treinamentos - modelo TF-IDF+MLP:

- Processos realizados utilizando a biblioteca TfidfVectorizer:
- Tokenização;
- Adição ngrams_range=1,2;
- min_df=8;
- max_df=0.3;
- Steaming;
- Remoção de stop words;
- Limitação de features: max_features=5.000
- Foram limitadas a quantidade de instâncias a serem treinadas, devido a sobrecarga de memória em vários momentos durante o treinamento;
- Treinamentos - modelo TF-IDF+MLP foram realizados até o momento com 16.704 instâncias;

5. Treinamento dos modelos - Transformers

Exemplo de sequência de variações de hiperparâmetros para cada modelo de Transformers:

❖ **Variação de hiperparâmetros:**

Dropout 0, Learning Rate: $2.5e-5$

Dropout 0, Learning Rate: $5e-5$

Dropout 10%, Learning Rate: $2.5e-5$

Dropout 10%, Learning Rate: $5e-5$

❖ **Seguindo a mesma lógica até finalizar as variações:**

- 2 sequências de variações (2 variações de dropout)
- Cada variação com 2 treinamentos (2 variações de learning rate)
- Para cada um dos 7 modelos;
- 28 treinamentos experimentais para cada uma das bases;
- Totalizando 56 treinamentos abrangendo as duas bases;

5. Treinamento dos modelos - Transformers

Modelos Transformers

Identificador	Modelo	Multilingue	Dimensões do Embedding
Ro_B	XML-Roberta-base	sim	768
Ro_L	XML-Roberta-large	sim	1024
Bert_B	Bert-base-multilingual-cased	sim	768

Modelos Transformers + FB

Modelo + FB	Modelo	Multilingue	Dimensões do Embedding
TML Pequeno + FB1 + MLP	XML-Roberta-base	sim	768
TML Pequeno + FB2 + MLP	XML-Roberta-base	sim	768
TLM Grande + FB1 + MLP	XML-Roberta-large	sim	1024
TLM Grande + FB2 + MLP	XML-Roberta-large	sim	1024
TLM Pequeno + Finetuning	XML-Roberta-base	sim	768
TLM Grande + Finetuning	XML-Roberta-large	sim	1024
TLM Multilingue	Bert-base-multilingual-cased	sim	768

5. Treinamento dos modelos - Embeddings - FB

FB1: Concatena Média, Min, e Max:

```
# Feature based 1 - Concatena Média, Min e Max  
# Calculando valores estatísticos:  
mean_values = np.mean(embeddings, axis=1)  
min_values = np.min(embeddings, axis=1)  
max_values = np.max(embeddings, axis=1)  
  
# Conferindo dimensões das matrizes dos valores estatísticos  
print(mean_values.shape)  
print(min_values.shape)  
print(max_values.shape)  
  
(133632, 768)  
(133632, 768)  
(133632, 768)
```

FB2: Concatena CLS, Média de todos, e Std:

```
# Feature based 2 - Concatena CLS, Med de todos, Std  
# Calculando valores estatísticos:  
cls_values = embeddings_narray[:,0,:]  
mean_values = np.mean(embeddings_narray, axis=1)  
std_values = np.std(embeddings_narray, axis=1)  
  
print(cls_values.shape)  
print(mean_values.shape)  
print(std_values.shape)  
  
(133632, 768)  
(133632, 768)  
(133632, 768)
```

6. Pós Treinamento - Testes Estatísticos

❖ Após a finalização de cada treinamento:

- Para cada treinamento, os dados de entrada foram divididos em 10 folds;
- Para cada fold foram extraídas métricas F1 Score e Acurácia;
- No final de cada treinamento foram geradas as médias de cada métrica dos respectivos folds;
- **Posteriormente utilizamos um algoritmo que desenvolvemos que tem a seguinte tarefa de estimar:**
 - Dada como entrada as médias das métricas F1 Score e Acurácia dos folds de cada modelo;
 - O algoritmo retorna:
 - A melhor média das métricas dos treinamentos;
 - Desvio padrão da F1 Score e Acurácia da melhor média das métricas;
 - Índice do melhor treinamento (treinamento com a melhor média);

6. Pós Treinamento - Testes Estatísticos

❖ **A partir dos resultados destes dados estatísticos geramos:**

- Tabela com melhores resultados dos treinamentos das respectivas bases;
- Tabela com análises estatísticas;
- Gráficos das métricas visando auxiliar em análises e comparações entre os modelos;
- Gráficos globais abrangendo os resultados dos treinamentos, para auxiliar em análises e comparações entre modelos e bases;

7. Análises Estatísticas - Melhores Modelos - Embeddings Estáticos

Modelo	Base de Dados	Hiperparametros	Processo	F1-Score	Acurácia
TFIDF+ MLP	Buscapé	Dropout = 0.05, Learning rate = 5e-3	Treino	0.99 ± 0.02	0.95 ± 0.02
			Teste	0.83 ± 0.01	0.83 ± 0.01
	B2W	Dropout = 0.05, Learning rate = 5e-4	Treino	0.96 ± 0.00	0.94 ± 0.01
			Teste	0.93 ± 0.01	0.93 ± 0.01
GloVe + MLP	Buscapé	Dropout = 0.05, Learning rate = 5e-4	Treino	0.80 ± 0.01	0.78 ± 0.01
			Teste	0.79 ± 0.01	0.79 ± 0.01
	B2W	Dropout = 0.05, Learning rate = 1e-3	Treino	0.87 ± 0.01	0.86 ± 0.01
			Teste	0.87 ± 0.01	0.86 ± 0.01
FastText + MLP	Buscapé	Dropout = 0.05, Learning rate = 5e-4	Treino	0.87 ± 0.01	0.86 ± 0.01
			Teste	0.87 ± 0.01	0.86 ± 0.01
	B2W	Dropout = 0.05, Learning rate = 1e-3	Treino	0.93 ± 0.01	0.92 ± 0.01
			Teste	0.92 ± 0.01	0.92 ± 0.01
Fast Text + Fine Tuning	Buscapé	Dropout = 0.05, Learning rate = 5e-3	Treino	0.88 ± 0.01	0.87 ± 0.01
			Teste	0.88 ± 0.01	0.88 ± 0.01
	B2W	Dropout = 0.05, Learning rate = 1e-3	Treino	0.93 ± 0.01	0.93 ± 0.01
			Teste	0.93 ± 0.01	0.93 ± 0.01
GloVe + Fine Tuning	Buscapé	Dropout = 0.05, Learning rate = 1e-3	Treino	0.81 ± 0.01	0.80 ± 0.01
			Teste	0.80 ± 0.01	0.80 ± 0.01
	B2W	Dropout = 0.05, Learning rate = 1e-3	Treino	0.87 ± 0.01	0.87 ± 0.01
			Teste	0.87 ± 0.01	0.87 ± 0.01

7. Análises Estatísticas - Melhores Modelos - Transformers

TLM Pequeno + FB1 + MLP	Buscapé	Dropout = 0, Learning rate = 2.5e-5	Treino	0.86 ± 0.01	0.86 ± 0.01
			Teste	0.86 ± 0.01	0.86 ± 0.01
	B2W	Dropout = 0, Learning rate = 5e-5	Treino	0.93 ± 0.00	0.93 ± 0.00
			Teste	0.93 ± 0.00	0.93 ± 0.00
TLM Pequeno + FB2 + MLP	Buscapé	Dropout = 0.10, Learning rate = 5e-5	Treino	0.85 ± 0.01	0.85 ± 0.01
			Teste	0.87 ± 0.01	0.87 ± 0.01
	B2W	Dropout = 0, Learning rate = 5e-5	Treino	0.93 ± 0.01	0.93 ± 0.01
			Teste	0.93 ± 0.01	0.93 ± 0.01
TLM Grande + FB1 + MLP	Buscapé	Dropout = 0.10, Learning rate = 5e-5	Treino	0.88 ± 0.01	0.87 ± 0.01
			Teste	0.88 ± 0.01	0.87 ± 0.01
	B2W	Dropout = 0, Learning rate = 5e-5	Treino	1.00 ± 0.01	0.93 ± 0.01
			Teste	1.00 ± 0.01	0.93 ± 0.01
TLM Grande + FB2 + MLP	Buscapé	Dropout = 0, Learning rate = 2.5e-5	Treino	0.91 ± 0.01	0.90 ± 0.02
			Teste	0.91 ± 0.01	0.90 ± 0.02
	B2W	Dropout = 0.10, Learning rate = 2.5e-5	Treino	0.94 ± 0.00	0.93 ± 0.01
			Teste	0.94 ± 0.00	0.93 ± 0.01
TLM Pequeno + FineTuning	Buscapé	Dropout = 0.10, Learning rate = 5e-5	Treino	0.90 ± 0.00	0.90 ± 0.01
			Teste	0.90 ± 0.00	0.90 ± 0.01
	B2W	Dropout = 0.10, Learning rate = 5e-5	Treino	0.90 ± 0.00	0.90 ± 0.01
			Teste	0.90 ± 0.00	0.90 ± 0.01
TLM Grande + FineTuning	Buscapé	Dropout = 0.10, Learning rate = 5e-5	Treino	0.89 ± 0.00	0.88 ± 0.00
			Teste	0.89 ± 0.00	0.88 ± 0.00
	B2W	Dropout = 0.10, Learning rate = 5e-5	Treino	1.00 ± 0.00	0.94 ± 0.00
			Teste	1.00 ± 0.00	0.94 ± 0.00
TLM Pequeno Multilingue	Buscapé	Dropout = 0.10, Learning rate = 5e-5	Treino	0.85 ± 0.01	0.84 ± 0.01
			Teste	0.85 ± 0.01	0.84 ± 0.01
	B2W	Dropout = 0.10, Learning rate = 5e-5	Treino	0.84 ± 0.01	0.83 ± 0.01
			Teste	0.84 ± 0.01	0.83 ± 0.01

8. Questões

1. Modelos com maior quantidade de parâmetros melhoraram a predição? Quando isso não ocorreu? Mostre os resultados.

- ❖ Modelos com maior quantidade de parâmetros não melhoraram a predição;
- ❖ No caso citado a seguir, demonstra um modelo com maior quantidade de parâmetros em relação a outro modelo, e não obteve melhora na predição:

Modelo	Quantidade Parâmetros	Acurácia Teste	Base
TFIDF + MLP	117.193	0.83 ± 0.01	buscapé
FastText + Fine Tuning	7.249	0.88 ± 0.01	buscapé

- ❖ Exemplo do modelo **TF-IDF + MLP**, teve **117.193** parâmetros e teve uma acurácia na predição de 0.83 na base buscapé;
- ❖ Enquanto o modelo **FastText + Fine Tuning** teve menor quantidade de parâmetros, **7.249** e melhor desempenho na predição com uma acurácia de 0.88 na base buscapé;

8. Questões

- 1. Modelos com maior quantidade de parâmetros melhoraram a predição? Quando isso não ocorreu? Mostre os resultados.**

Sumarização da rede : TFIDF + MLP

Layer (type)	Output Shape	Param #
dense_18 (Dense)	(None, 24)	117168
dropout_9 (Dropout)	(None, 24)	0
dense_19 (Dense)	(None, 1)	25
Total params: 117,193		
Trainable params: 117,193		
Non-trainable params: 0		

Sumarização da rede : FastText + Fine Tuning

Layer (type)	Output Shape	Param #
dense_2 (Dense)	(None, 24)	7224
dropout_1 (Dropout)	(None, 24)	0
dense_3 (Dense)	(None, 1)	25
Total params: 7,249		
Trainable params: 7,249		
Non-trainable params: 0		

8. Questões

1. **Modelos com maior quantidade de parâmetros melhoraram a predição? Quando isso não ocorreu? Mostre os resultados.**
 - ❖ Na base B2W os dois modelos empataram com 0.93 de acurácia;
 - ❖ Portanto consideramos:
 - Modelo FastText + Fine Tuning com menor quantidade de parâmetros teve melhor desempenho na predição;
 - Modelo TF-IDF + MLP , com maior quantidade de parâmetros teve desempenho inferior;

8. Questões

1. Modelos com maior quantidade de parâmetros melhoraram a predição? Quando isso não ocorreu? Mostre os resultados.

Tabela Quantidade de Parâmetros: Métrica Acurácia na Predição:

Modelo	Total de Parâmetros	Acurácia - Base Buscapé	Acurácia - Base B2W
tfidf-mlp	117.193	0.83 ± 0.01	0.93 ± 0.01
gloVe-mlp	1.075	0.79 ± 0.01	0.86 ± 0.01
gloVe-finetuning	1.075	0.80 ± 0.01	0.87 ± 0.01
fasttext-mlp	7.249	0.86 ± 0.01	0.92 ± 0.01
fasttext-finetuning	7.249	0.88 ± 0.01	0.93 ± 0.01
tlm-pequeno-fb1-mlp	58.417	0.86 ± 0.01	0.93 ± 0.00
tlm-pequeno-fb2-mlp	58.417	0.87 ± 0.01	0.93 ± 0.01
tlm-grande-fb1-mlp	24.649	0.87 ± 0.01	0.93 ± 0.01
tlm-grande-fb2-mlp	24.673	0.90 ± 0.02	0.93 ± 0.01
tlm-pequeno-finetuning	18.505	0.90 ± 0.01	0.90 ± 0.01
tlm-grande-finetuning	24.649	0.88 ± 0.00	0.94 ± 0.00
tlm-pequeno-multilingual	18.505	0.84 ± 0.01	0.84 ± 0.01

8. Questões

2. Quais modelos foram os melhores, piores, e quais tiveram empate?

Melhor Modelo		
Base	Modelo	Acurácia Teste
B2W	TLM Grande + Fine Tuning	0.94

Piores Modelos		
Base	Modelo	Acurácia Teste
B2W	TLM Pequeno Multilingue	0.83
Buscapé	GloVe+MLP	0.79

8. Questões

2. Quais modelos foram os melhores, e piores, e quais tiveram empate?

Modelos Empatados		
Base	Modelo	Acurácia Teste
Buscapé	FastText + Fine Tuning	0.88
Buscapé	TLM Grande + Fine Tuning	0.88
B2W	TFIDF + MLP	0.93
B2W	FastText + Fine Tuning	0.93
B2W	TLM Pequeno + FB1 + MLP	0.93
B2W	TLM Pequeno + FB2 + MLP	0.93

8. Questões

- 3. Ocorreu impacto positivo na predição considerando modelos multilíngue em relação aos modelos em português?**
- ❖ Objetivando a análise dos modelos multilíngue em relação aos modelos português:
 - Separamos os melhores resultados dos modelos de transformers pré-treinados em português;
 - Separamos os melhores resultados do modelo multilíngue TLM pequeno multilíngue BERT base;
 - Em seguida realizamos tentativas de comparações dos resultados:

8. Questões

3. Ocorreu impacto positivo na predição considerando modelos multilíngue em relação aos modelos em português?

Modelos Multilíngue			
Base	Modelo	Acurácia Treino	Acurácia Teste
Buscape B2W	TLM Pequeno Multilíngue	0.84 ± 0.01	0.84 ± 0.01
	TLM Pequeno Multilíngue	0.83 ± 0.01	0.83 ± 0.01

Modelos Português			
Base	Modelo	Acurácia Treino	Acurácia Teste
Buscape	TLM Grande + FB2 + MLP	0.90 ± 0.02	0.93 ± 0.01
	TLM Pequeno + FineTuning	0.90 ± 0.01	0.90 ± 0.01
	TLM Grande + FineTuning	0.88 ± 0.00	0.88 ± 0.00
	TLM Grande + FB1 + MLP	0.87 ± 0.01	0.87 ± 0.01
	TLM Pequeno + FB2 + MLP	0.85 ± 0.01	0.87 ± 0.01
	TLM Pequeno + FB1 + MLP	0.86 ± 0.01	0.86 ± 0.01
B2W	TLM Grande + FineTuning	0.94 ± 0.00	0.94 ± 0.00
	TLM Pequeno + FB1 + MLP	0.93 ± 0.00	0.93 ± 0.00
	TLM Pequeno + FB2 + MLP	0.93 ± 0.01	0.93 ± 0.01
	TLM Grande + FB1 + MLP	0.93 ± 0.01	0.93 ± 0.01
	TLM Grande + FB2 + MLP	0.93 ± 0.01	0.93 ± 0.01
	TLM Pequeno + FineTuning	0.90 ± 0.01	0.90 ± 0.01

- Observamos que modelos multilíngue treinados em nossos experimentos não obtiveram melhores performances em relação a modelos TLM em Português;
- Enquanto o melhor modelo multilíngue (TLM Pequeno Multilíngue - base buscapé) alcançou acurácia no teste de 0.84;
- O Modelo TLM Grande + Fine Tuning, Português, base B2W, superou com acurácia no teste de 0.94;
- Portanto concluímos que não ocorreram impacto positivo na predição considerando modelos multilíngue em relação ao modelos em português;

8. Questões

4. Qual o ranking dos modelos de agregação dos embeddings de Feature Based?

- ❖ Base buscapé - Modelo TLM Grande+FB2+MLP teve melhor desempenho com acurácia de 0.90;
- ❖ Base B2W empate entre todos os modelos de Feature Based com acurácia de 0.93;

Ranking de modelos FB1 e FB2			
Base buscape	Acurácia	Base B2W	Acurácia
TLM Grande + FB2 + MLP	0.90	TLM Pequeno + FB1 + MLP	0.93
TLM Grande + FB1 + MLP	0.87	TLM Pequeno + FB2 + MLP	0.93
TLM Pequeno + FB1 + MLP	0.86	TLM Grande + FB1 + MLP	0.93
TLM Pequeno + FB2 + MLP	0.85	TLM Grande + FB2 + MLP	0.93

8. Questões

5. Em quais modelos o uso de contexto melhorou a predição e em quais Embeddings estáticos foram competitivos?

❖ **Uso de contexto no modelo TFIDF+MLP:**

- **Pré-processamento na tentativa de melhorar o contexto do corpus;**
- Ajustes de hiperparâmetros da função TfidfVectorizer, também fazendo parte do processo de pré-processamento:
- min_df=8: sugerido no projeto a retirada de palavras frequência menor que 5, porém na tentativa de aumentar a quantidade de instâncias durante o treinamento, decidimos retirar palavras com frequência menor que 8;
- Adição do hiperparâmetro max_df=0.3 (30%): ignora os termos que tenham frequência de documento estritamente superior ao limite determinado;
- **Ajustes realizados visando:**
- Melhora do contexto;
- Redução da dimensionalidade;
- Melhora na alocação de memória durante o treinamento do modelo, evitando travamentos;
- Melhora na predição do modelo;
- **Resultados após os ajustes no contexto:**
- Quantidade de instâncias no treinamento aumentou de 8.340 para 16.680;
- Melhora na predição do modelo;

8. Questões

5. Em quais modelos o uso de contexto melhorou a predição e em quais Embeddings estáticos foram competitivos?
- Modelos de embeddings estáticos que foram competitivos:

Predições Competitivas		
Base	Modelo	Acurácia
Buscapé	GloVe+Fine Tuning	0.80
	GloVe+MLP	0.79
B2W	TFIDF+MLP	0.93
	FastText+MLP	0.92

8. Questões

6. Em quais casos o fino ajuste melhorou a predição em relação aos resultados com feature based?

❖ Base buscapé:

- Modelos **TLM Grande+FineTuning e FastText+FineTuning** melhoraram sua predição com uma **acurácia de 0.88** em relação ao **Modelo TLM Grande+FB1+MLP** com **acurácia de 0.87**;
- Modelos **TLM Pequeno+FineTuning e TLM Grande+FB2+MLP** empataram com **acurácia de 0.90**;

❖ Base B2W:

- Modelo **TLM Grande+FineTuning** melhorou sua predição com **acurácia de 0.94** em relação a **todos modelos FB1 e FB2**, os mesmos obtiveram um empate em 0.93 de acurácia;

Ranking de modelos FB1 e FB2

Base buscape	Acurácia	Base B2W	Acurácia
TLM Grande + FB2 + MLP	0.90	TLM Pequeno + FB1 + MLP	0.93
TLM Grande + FB1 + MLP	0.87	TLM Pequeno + FB2 + MLP	0.93
TLM Pequeno + FB1 + MLP	0.86	TLM Grande + FB1 + MLP	0.93
TLM Pequeno + FB2 + MLP	0.85	TLM Grande + FB2 + MLP	0.93

Ranking de modelos Fine Tuning

Base buscape	Acurácia	Base B2W	Acurácia
TLM Pequeno + FineTuning	0.90	TLM Grande + FineTuning	0.94
TLM Grande + FineTuning	0.88	Fast Text + Fine Tuning	0.93
Fast Text + Fine Tuning	0.88	TLM Pequeno + FineTuning	0.90
GloVe + Fine Tuning	0.80	GloVe + Fine Tuning	0.87

8. Questões

7. Compare os resultados do seu trabalho com os do artigo.

- ❖ Tentando comparar os resultados do artigo com projeto, associamos modelos do artigo com modelos utilizados no projeto.
- ❖ Na tabela a seguir é demonstrado esta relação:

Relacionamento - Modelos Artigo - Modelos Projeto NLP 2023

Modelos Artigo: Embedding generation for text classification of Brazilian Portuguese user reviews: from bag-of-words to transformers	Modelos do Projeto NLP 2023
BoW	TF-IDF+MLP
CNN	GloVe+MLP
	FastText+MLP
	GloVe+Fine Tuning
LSTM	FastText+Fine Tuning
	TLM Peq. + FB1 + MLP
	TLM Peq. + FB2 + MLP
FB TLM	TLM Peq. + Fine Tuning
	TLM Grande + Fine Tuning
	TLM Multilingue

8. Questões

Questão 7 - Comparativos - Projeto e Artigo:

- ❖ Detalhamos na tabela a seguir as relações e os respectivos valores de métricas de acurácia e f1 score do artigo e do projeto, para fins de comparação:

Comparativos de Resultados do Artigo e Experimentos do Projeto						
Resultados do Artigo: Embedding generation for text classification of Brazilian Portuguese user reviews: from bag-of-words to transformers				Resultados de Experimentos do Projeto		
Métrica	Modelo	Buscape	B2W	Modelo	Buscape	B2W
Acurácia	BoW	0.948 ± 0.02	0.940 ± 0.03	TF-IDF+MLP	0.83 ± 0.01	0.93 ± 0.01
	CNN	0.957 ± 0.02	0.947 ± 0.06	GloVe+MLP	0.79 ± 0.01	0.86 ± 0.01
	LSTM	0.955 ± 0.02	0.944 ± 0.01	GloVe+Fine Tuning	0.80 ± 0.01	0.87 ± 0.01
	CNN	0.957 ± 0.02	0.947 ± 0.06	FastText+MLP	0.86 ± 0.01	0.92 ± 0.01
	LSTM	0.955 ± 0.02	0.944 ± 0.01	FastText+Fine Tuning	0.88 ± 0.01	0.93 ± 0.01
	FB TLM	0.948 ± 0.02	0.940 ± 0.03	TLM Peq. + FB1 + MLP	0.86 ± 0.01	0.93 ± 0.00
	FB TLM	0.948 ± 0.02	0.940 ± 0.03	TLM Peq. + FB2 + MLP	0.87 ± 0.01	0.93 ± 0.01
	FT TLM	0.978 ± 0.01	0.978 ± 0.01	TLM Peq. + Fine Tuning	0.90 ± 0.01	0.90 ± 0.01
	FT TLM	0.978 ± 0.01	0.978 ± 0.01	TLM Grande + Fine Tuning	0.88 ± 0.00	0.94 ± 0.00
	FT TLM	0.978 ± 0.01	0.978 ± 0.01	TLM Multilingue	0.84 ± 0.01	0.84 ± 0.01
F1-Score	BoW	0.972 ± 0.01	0.957 ± 0.02	TF-IDF+MLP	0.83 ± 0.01	0.93 ± 0.01
	CNN	0.977 ± 0.01	0.962 ± 0.04	GloVe+MLP	0.79 ± 0.01	0.87 ± 0.01
	LSTM	0.976 ± 0.01	0.960 ± 0.06	GloVe+Fine Tuning	0.80 ± 0.01	0.87 ± 0.01
	CNN	0.977 ± 0.01	0.962 ± 0.04	FastText+MLP	0.87 ± 0.01	0.92 ± 0.01
	LSTM	0.976 ± 0.01	0.960 ± 0.06	FastText+Fine Tuning	0.88 ± 0.01	0.93 ± 0.01
	FB TLM	0.976 ± 0.01	0.972 ± 0.01	TLM Peq. + FB1 + MLP	0.86 ± 0.01	0.93 ± 0.00
	FB TLM	0.976 ± 0.01	0.972 ± 0.01	TLM Peq. + FB2 + MLP	0.87 ± 0.01	0.93 ± 0.01
	FT TLM	0.978 ± 0.01	0.978 ± 0.01	TLM Peq. + Fine Tuning	0.90 ± 0.00	0.90 ± 0.00
	FT TLM	0.978 ± 0.01	0.978 ± 0.01	TLM Grande + Fine Tuning	0.89 ± 0.00	1.00 ± 0.00
	FT TLM	0.978 ± 0.01	0.978 ± 0.01	TLM Multilingue	0.84 ± 0.01	0.84 ± 0.01

8. Questões

8. Como a representação pré-treinada foi construída? Descreva métodos de treinamento, base de dados utilizadas, e dimensões do modelo transformers.

- Os modelos de representações pré treinadas designados para nosso grupo foram:
 - xml-roberta-base;
 - xml-roberta-large;
 - bert-large-multilíngue (substituído: bert-base-multilingual-cased);

Representações Pré-Treinadas

Modelo	Detalhamento das Siglas	Dados Filtrados	Linguagens	Dimensões	Tamanho do Vocabulário
xml-roberta-base	A Robustly Optimized BERT Pre Training Approach	2.5TB Common Crawl	100	768	250.002
xml-roberta-large	A Robustly Optimized BERT Pre Training Approach	2.5TB Common Crawl	100	1024	250.002
bert-base-multilingual-cased	Bidirectional Encoder Representations from Transformers	Largest Wikipedia (MLM)	104	768	119.547

- Detalhamento dos tipos de tokens;
- Tokens dos transformers xml-roberta-base e large são iguais;
- Em seguida tokens bert-base-multilingual-cased;

```
tokenizer.special_tokens_map

{'bos_token': '<s>',
 'eos_token': '</s>',
 'unk_token': '<unk>',
 'sep_token': '</s>',
 'pad_token': '<pad>',
 'cls_token': '<s>',
 'mask_token': '<mask>'}

print('bos_token_id <s>:', tokenizer.bos_token_id)
print('eos_token_id </s>:', tokenizer.eos_token_id)
print('sep_token_id </s>:', tokenizer.sep_token_id)
print('pad_token_id <pad>:', tokenizer.pad_token_id)

bos_token_id <s>: 0
eos_token_id </s>: 2
sep_token_id </s>: 2
pad_token_id <pad>: 1
```

```
[6] tokenizer.special_tokens_map

{'unk_token': '[UNK]',
 'sep_token': '[SEP]',
 'pad_token': '[PAD]',
 'cls_token': '[CLS]',
 'mask_token': '[MASK]'}

print('unk_token_id <s>:', tokenizer.unk_token_id)
print('sep_token_id </s>:', tokenizer.sep_token_id)
print('pad_token_id </s>:', tokenizer.pad_token_id)
print('cls_token_id <pad>:', tokenizer.cls_token_id)
print('mask_token_id <pad>:', tokenizer.mask_token_id)

unk_token_id <s>: 100
sep_token_id </s>: 102
pad_token_id </s>: 0
cls_token_id <pad>: 101
mask_token_id <pad>: 103
```

8. Questões

8. Como a representação pré-treinada foi construída? Descreva métodos de treinamento, base de dados utilizadas, e dimensões do modelo transformers.

- Modelo RoBERTa (A Robustly Optimized BERT Pre Training Approach) - xml-roberta-base:
 - Construído utilizando a arquitetura Transformer, que é um tipo de rede neural projetada para processar sequências de dados, como por exemplo texto;
 - Baseada no Google's BERT;
 - Release do modelo publicada em 2018;
 - Algumas características complementares:
 - Remoção do objetivo de pré-treinamento da próxima frase;
 - Treinamento com mini lotes e taxas de aprendizagem muito maiores;

xml-roberta-base	
Layers	12
Hidden size	768
Parameters	110 M
Attention heads	12

8. Questões

8. Como a representação pré-treinada foi construída? Descreva métodos de treinamento, base de dados utilizadas, e dimensões do modelo transformers.

- Modelo RoBERTa Large (Large-sized-model):
 - Pré-treinado em 2,5TB de dados CommonCrawl filtrados;
 - Common Crawl é uma organização sem fins lucrativos fundada em 2007;
 - Dados filtrados da web, cerca de 240 bilhões de páginas no decorrer de 16 anos;
 - 82% dos tokens brutos utilizados para treinar o GPT-3;
 - Roberta Large é pré-treinado apenas nos textos brutos, sem nenhum tipo de rótulo humano;
 - Por esse motivo ele pode utilizar muitos dados disponíveis publicamente;
 - Contém processo automático para gerar entrada e rótulos a partir desses textos;

xml-roberta-large	
Layers	24
Hidden size	1.024
Parameters	355 M
Attention heads	16

8. Questões

8. Como a representação pré-treinada foi construída? Descreva métodos de treinamento, base de dados utilizadas, e dimensões do modelo transformers.

- Modelo BERT base multilingual cased:
 - Modelo pré-treinado nos 104 principais idiomas;
 - Utiliza a técnica (MLM) Masked Language Modeling:
 - Pegando uma frase, o modelo máscara aleatoriamente 15% das palavras de entrada;
 - Em seguida executa toda a frase mascarada no modelo;
 - Em seguida tenta prever as palavras mascaradas;
- Diferencia letras maiúsculas de minúsculas;

bert-base-multilingual-cased	
Layers	12
Hidden size	768
Parameters	110 M
Attention heads	12

9. Referências

link: <https://huggingface.co/xlm-roberta-base>

link: <https://huggingface.co/xlm-roberta-large>

link: <https://huggingface.co/bert-base-multilingual-cased>

link:

https://github.com/jalammar/jalammar.github.io/blob/master/notebooks/bert/A_Visual_Notebook_to_Using_BERT_for_the_First_Time.ipynb

link: <https://fasttext.cc/docs/en/pretrained-vectors.html>

link: <https://nlp.stanford.edu/projects/glove/>

link bases de dados: <https://drive.google.com/drive/folders/15x4N8TM-veLYBG6-iiNAfGILFXoCBqxe>

link repositório do projeto: <https://github.com/brugnaro2022/nlp>