Universidade Federal do Piauí

Departamento de Ciência da Computação

Disciplina: Processamento de Linguagem Natural

Professor: Raimundo Santos Moura

Classificação de Planos Governamentais do Estado do Piauí

Cayo Cesar L. M. Pires Cardoso

20209051311

Icaro Gabryel de Araujo Silva

20209050584

13 de Janeiro de 2025

1. Introdução

A análise de propostas governamentais é uma atividade crucial para compreender as prioridades e intenções dos gestores públicos. No contexto do estado do Piauí, a classificação automatizada das propostas em eixos temáticos, como Saúde, Educação e Segurança, oferece insights que podem facilitar o acompanhamento, a transparência e a cobrança da execução dessas promessas.

Este projeto utiliza algoritmos de aprendizado de máquina, incluindo **Random Forest**, **MLP (Multilayer Perceptron)** e **BERT (Bidirectional Encoder Representations from Transformers)**, para classificar automaticamente as propostas. O objetivo é avaliar o desempenho dessas técnicas em um conjunto de dados real composto por propostas de governo das eleições estaduais de 2024 no Piauí.

A escolha dessas abordagens se baseia na eficiência comprovada de modelos tradicionais, como Random Forest e MLP, combinada com a robustez dos modelos baseados em transformers, como o BERT, amplamente utilizado no Processamento de Linguagem Natural (PLN). Por meio desse projeto, pretende-se construir um sistema confiável e interpretável para auxiliar a categorização automática das propostas governamentais, contribuindo para o avanço na área de inteligência artificial aplicada à análise política.

2. Descrição do Corpus

O corpus utilizado neste projeto é composto por um total de 1.170 propostas governamentais provenientes do estado do Piauí, especificamente das eleições de 2024. Essas propostas estão distribuídas em cinco eixos temáticos: Educação, Infraestrutura, Meio Ambiente, Saúde e Segurança. Cada eixo representa um conjunto de prioridades específicas detalhadas pelos candidatos em seus planos de governo.

A tabela 1 apresenta a distribuição quantitativa das propostas por eixo:

Eixo Temático	Quantidade de Propostas	Proporção (%)
Educação	257	21,97%
Infraestrutura	240	20,51%
Meio Ambiente	231	19,74%
Saúde	218	18,63%
Segurança	228	19,49%
Total	1170	100%

As propostas foram inicialmente extraídas de documentos oficiais em formato PDF e, em seguida, pré-processadas para remoção de elementos indesejados, como pontuação e palavras irrelevantes (**stopwords**). Além disso, as propostas foram organizadas em um formato

estruturado, contendo as colunas “Proposta” e “Eixo”, onde cada linha representa uma proposta classificada no respectivo eixo temático.

Um exemplo de proposta incluída no corpus pode ser observado a seguir:

- **Proposta**: “Garantir a construção e reforma de escolas em zonas rurais.”
- **Eixo Temático**: Educação.
- **Modelo de Proposta no Corpus: “**Garantir a construção e reforma de escolas em zonas rurais, Educação**”**

Esse corpus foi preparado especificamente para experimentos com modelos de aprendizado supervisionado e passou por etapas rigorosas de validação e limpeza. Cada uma das propostas possui apenas um eixo como rótulo, caracterizando um problema de classificação multiclasse.

3. Metodologia

A metodologia utilizada neste projeto envolve o uso de técnicas de **Processamento de Linguagem Natural (PLN)** e aprendizado de máquina para a classificação das propostas governamentais em eixos temáticos. O processo foi dividido em cinco etapas principais: **coleta e organização dos dados**, **pré-processamento**, **definição dos modelos**, **treinamento e validação**, e **avaliação e comparação de resultados**.

3.1 Coleta e Organização dos Dados

As propostas governamentais foram coletadas de documentos oficiais disponibilizados no formato PDF e organizadas em uma estrutura tabular, contendo duas colunas: `Proposta` (o texto descritivo da proposta) e `Eixo` (o rótulo temático da proposta). Esses dados foram salvos em um arquivo no formato CSV e contêm um total de 1.170 propostas distribuídas entre cinco eixos: Educação, Infraestrutura, Meio Ambiente, Saúde e Segurança.

3.2 Pré-processamento

Antes de realizar a classificação, foi aplicado um conjunto de técnicas de pré-processamento aos textos das propostas, incluindo:

- **Remoção de caracteres especiais**: Eliminação de pontuação e símbolos que não agregam valor semântico ao texto.
- **Tokenização**: Segmentação dos textos em unidades léxicas, como palavras ou frases.
- **Normalização**: Conversão de palavras para letras minúsculas para garantir uniformidade.
- **Remoção de stopwords**: Exclusão de palavras de alta frequência, mas sem relevância semântica (ex.: "de", "a", "o").

- **Transformação em vetores TF-IDF**: Utilizada para representar numericamente os textos no espaço vetorial, calculando a importância de cada termo em relação ao corpus.

No caso do modelo BERT, o pré-processamento seguiu a abordagem de tokenização específica fornecida pelo modelo pré-treinado, mantendo o contexto semântico original.

3.3 Definição dos Modelos

Foram utilizados três modelos de aprendizado supervisionado, cada um explorando características distintas na tarefa de classificação:

- **Random Forest**: Um modelo de ensemble learning baseado em múltiplas árvores de decisão. Esse modelo foi otimizado por meio da técnica de Grid Search, ajustando hiperparâmetros como número de árvores (**n_estimators**) e profundidade máxima (**max_depth**).
- **MLP (Multilayer Perceptron)**: Uma rede neural densa composta por camadas de neurônios interconectados. O MLP utiliza embeddings TF-IDF como entrada e aplica camadas de ativação (**ReLU**) e regularização (**Dropout**) para melhorar o desempenho e evitar o overfitting.
- **BERT (Bidirectional Encoder Representations from Transformers)**: Um modelo baseado em transformers, pré-treinado em grandes corpora de texto. O BERT foi ajustado para a tarefa de classificação multiclasse, com fine-tuning no corpus específico de propostas governamentais.

3.4 Treinamento e Validação

O dataset foi dividido em duas partes: 80% para treinamento e 20% para validação.

- **Random Forest e MLP**: Utilizaram-se vetores TF-IDF para representação das propostas e foram treinados com ênfase na otimização de métricas como acurácia e F1-score.
- **BERT**: Utilizou o método de fine-tuning, adaptando os pesos do modelo pré-treinado ao corpus de propostas. A tokenização e o treinamento foram realizados utilizando o framework Hugging Face Transformers.

3.5 Avaliação e Comparação de Resultados

Os três modelos foram avaliados no conjunto de validação utilizando métricas clássicas, como:

- **Acurácia**: Percentual de previsões corretas.
- **F1-Score**: Média harmônica entre precisão e recall, especialmente relevante para avaliar equilíbrio entre classes.
- **Matriz de Confusão**: Visualizou erros de classificação e a relação entre predições corretas e incorretas.

Além disso, foi analisada a **importância das features** no modelo Random Forest para entender quais termos contribuíram mais para a classificação. Para o modelo MLP e BERT, avaliou-se o impacto dos embeddings e a representatividade das palavras.

4. Implementação

A implementação do projeto foi estruturada em um pipeline de classificação, com etapas claramente definidas para **pré-processamento**, **treinamento dos modelos e avaliação** Cada modelo (Random Forest, MLP e BERT) teve sua configuração ajustada para atender às particularidades da tarefa de classificação das propostas governamentais.

4.1 Etapas do Pipeline de Classificação

4.1.1 Pré-processamento

As técnicas de pré-processamento aplicadas variaram de acordo com o modelo utilizado:

a) TF-IDF (Random Forest e MLP):

Transformação TF-IDF: As propostas foram convertidas em vetores numéricos, considerando a importância de cada palavra no texto (TF-IDF).

```
vectorizer = TfidfVectorizer(max_features=5000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_val_tfidf = vectorizer.transform(X_val)
```

- **Impacto:** Essa representação numérica foi usada como entrada nos modelos Random Forest e MLP.

b) Tokenização BERT:

O modelo BERT requer uma tokenização própria que preserva o contexto das palavras, realizada usando o BertTokenizer da biblioteca **Transformers**:

```
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")
train_encodings = tokenizer(list(X_train), truncation=True, padding="max_length", max_length=128,
return_tensors="pt")
val_encodings = tokenizer(list(X_val), truncation=True, padding="max_length", max_length=128,
return_tensors="pt")
```

4.1.2 Treinamento dos Modelos

Cada modelo foi configurado e otimizado com parâmetros específicos:

a) Random Forest:

```
    param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'criterion': ['gini', 'entropy']
}
grid_search = GridSearchCV(estimator=RandomForestClassifier(random_state=42),
                           param_grid=param_grid, cv=3, verbose=2, n_jobs=-1)
grid_search.fit(X_train_tfidf, y_train)
best_rf = grid_search.best_estimator_
```

Usamos **Grid Search** para ajustar hiperparâmetros como o número de árvores (n_estimators) e profundidade máxima (max_depth):O melhor modelo encontrado foi usado para prever os rótulos no conjunto de validação.

b) MLP (Multilayer Perceptron):

Configuramos uma rede neural com três camadas principais:

- Camada densa inicial com 512 neurônios e ativação ReLU.
- Camadas Dropout para evitar overfitting.
- Camada de saída com ativação softmax para classificação multiclasse:

```
model = Sequential([
    Dense(512, input_dim=X_train_tfidf.shape[1], activation='relu'),
    Dropout(0.5),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(len(label_encoder.classes_), activation='softmax')
])
model.compile(optimizer='adam', loss='categorical_crossentropy',
              metrics=['accuracy'])
history = model.fit(X_train_tfidf, y_train_one_hot, epochs=10, batch_size=32,
                    validation_data=(X_val_tfidf, y_val_one_hot))
```

c) BERT:

Realizamos o fine-tuning de um modelo pré-treinado bert-base-uncased, adaptando os pesos para as propostas governamentais:

```
model = BertForSequenceClassification.from_pretrained("bert-base-uncased",
num_labels=len(label_encoder.classes_))
training_args = TrainingArguments(
    output_dir="./results",
    evaluation_strategy="epoch",
    save_strategy="epoch",
    num_train_epochs=3,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    logging_dir="./logs",
    logging_steps=10,
    load_best_model_at_end=True
)
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    tokenizer=tokenizer
)
trainer.train()
```

4.1.3 Avaliação

A avaliação dos modelos foi realizada utilizando um conjunto de métricas que permitiram comparar a precisão, a consistência e a eficácia das classificações realizadas. A matriz de confusão foi empregada para visualizar os acertos e erros das previsões, destacando a relação entre as classificações corretas e incorretas para cada eixo temático. Essa abordagem foi fundamental para identificar padrões de confusão entre classes específicas, como a proximidade semântica entre eixos como Educação e Infraestrutura. Além disso, o relatório de classificação foi gerado para cada modelo, detalhando métricas como precisão, recall e F1-score. Essas métricas foram especialmente úteis para avaliar o equilíbrio do desempenho do modelo em classes desbalanceadas, como Saúde e Meio Ambiente.

Para o modelo Random Forest, foi realizada uma análise da importância das features, que revelou quais palavras foram mais relevantes na decisão final do modelo. Isso forneceu insights valiosos sobre os termos mais representativos de cada eixo temático. No caso dos modelos MLP e BERT, a avaliação considerou o impacto dos embeddings na performance geral, evidenciando a robustez do BERT para capturar relações contextuais mais complexas nos textos. Essas avaliações permitiram identificar os pontos fortes de cada modelo e os desafios associados à classificação automática das propostas governamentais.

5. Resultados e Análise

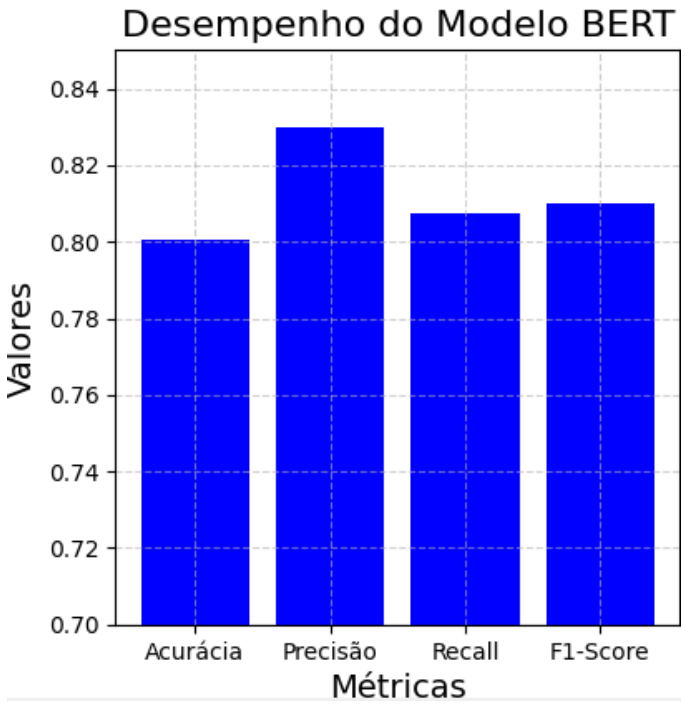
Este tópico apresenta os resultados obtidos pelos modelos BERT, Random Forest e MLP na tarefa de classificação das propostas governamentais. A avaliação foi realizada utilizando um conjunto de validação contendo 234 exemplos, considerando métricas como acurácia, precisão, recall e F1-score. Os principais resultados e comparações estão detalhados a seguir.

5.1 Desempenho dos Modelos

BERT

O modelo BERT, ajustado por meio de fine-tuning para a tarefa de classificação, alcançou uma acurácia de 80% no conjunto de validação. O F1-score médio foi de 79,79%, com uma precisão geral de 82,98% e um recall de 79,52%. O aumento nas métricas reforça a robustez do BERT para capturar padrões complexos em textos, especialmente em categorias que apresentam sobreposição temática.

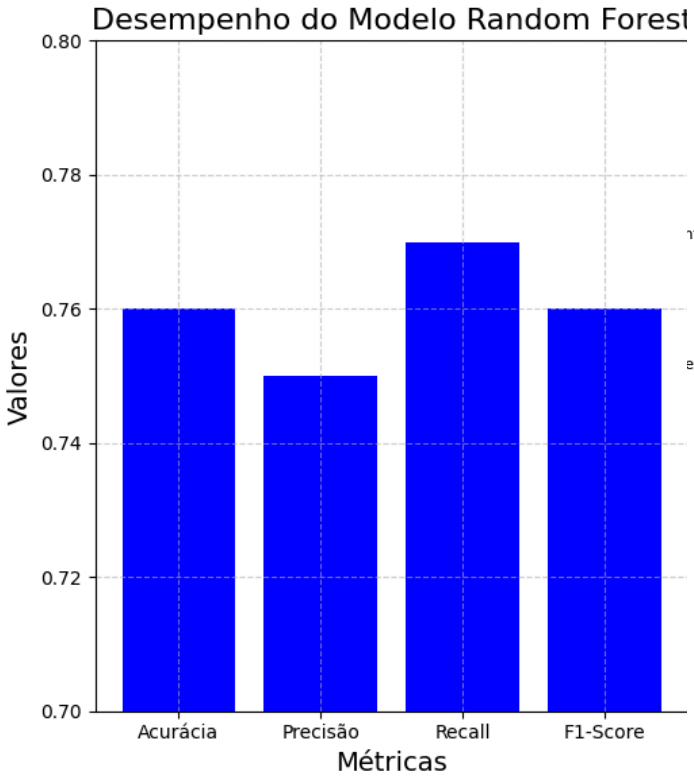
Na análise classe a classe, destaca-se a alta precisão no eixo Segurança (95%), mantendo consistência com um recall de 74%. O desempenho foi equilibrado nas demais categorias, com destaque para Meio Ambiente (82% de precisão e 79% de F1-score) e Educação (70% de precisão e 95% de recall). Embora exija maior capacidade computacional e maior tempo de treinamento, o modelo apresentou uma eficácia superior na generalização de padrões contextuais.

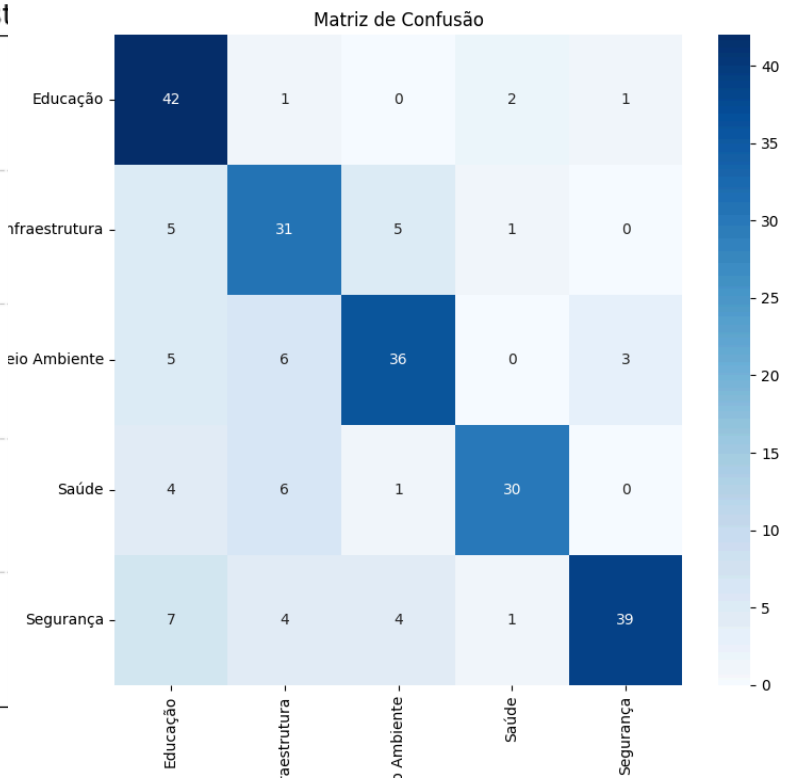


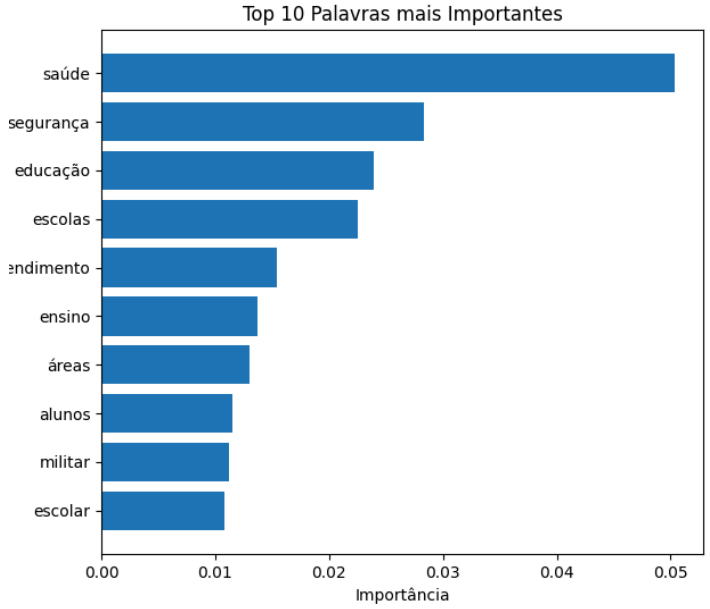
Random Forest

O modelo Random Forest apresentou um desempenho sólido, alcançando uma acurácia de 78%. Com um F1-score médio de 76% e precisão de 78%, esse modelo mostrou-se competitivo, sobretudo por sua eficiência computacional em comparação ao BERT.

Classe a classe, o Random Forest destacou-se com alta precisão para o eixo Meio Ambiente (83%), além de um bom equilíbrio em categorias como Infraestrutura (76% de precisão e 81% de recall) e Saúde (88% de precisão e 85% de recall). O modelo foi eficiente na identificação de termos específicos que diferenciam as classes, mas teve confusões pontuais em eixos semanticamente correlatos.



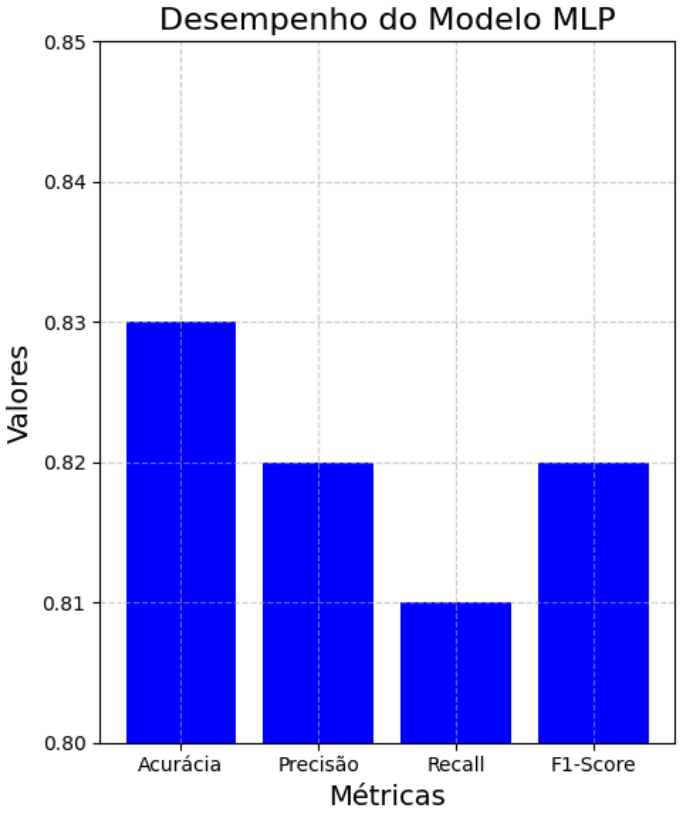


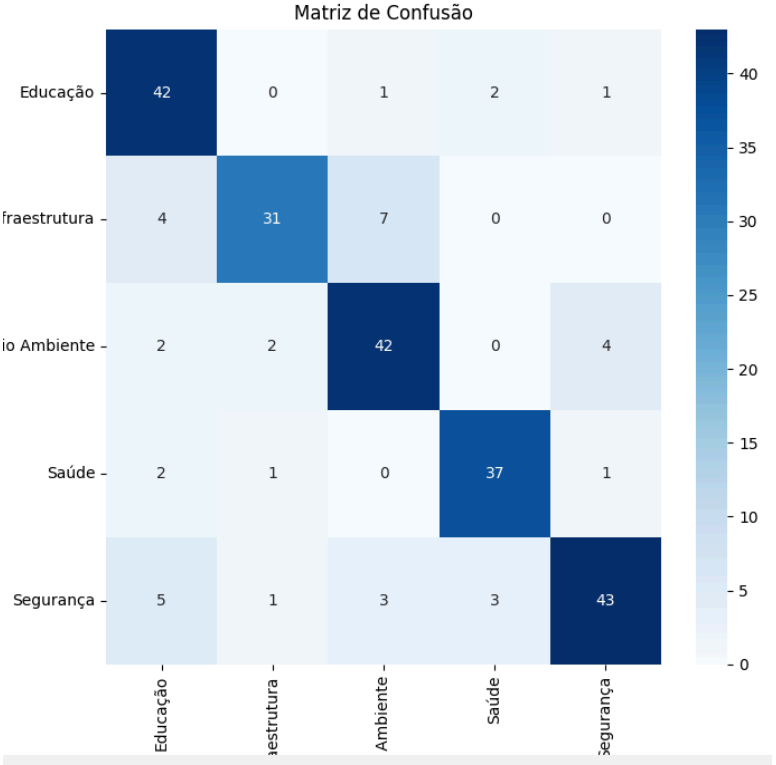


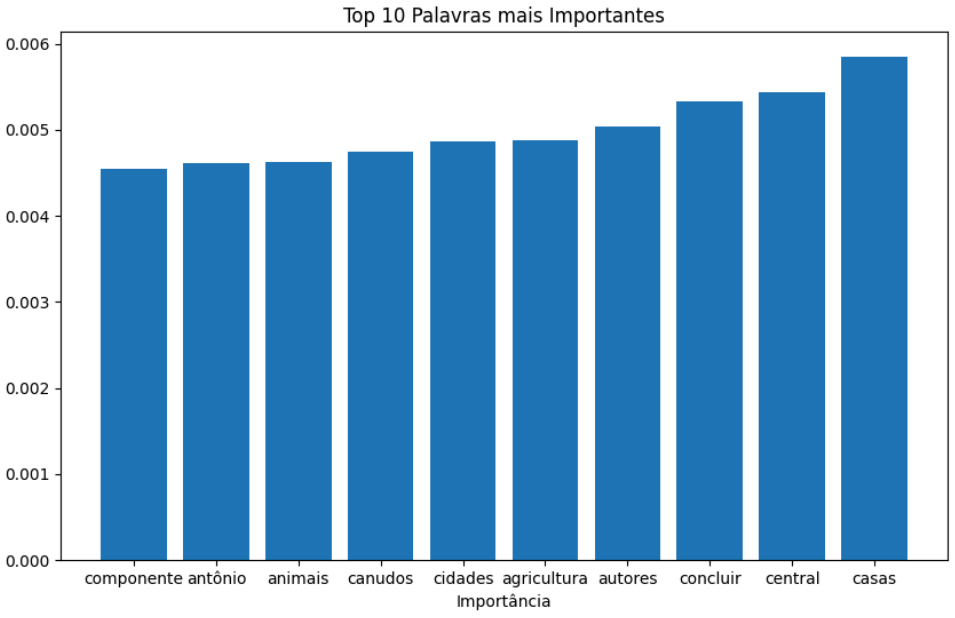
MLP

A rede neural MLP obteve o melhor desempenho geral, atingindo uma acurácia de 83%. O F1-score médio ponderado foi de 83%, com precisão média de 84% e recall médio de 84%, indicando um ótimo equilíbrio entre as métricas.

O modelo demonstrou excelente desempenho na categoria Saúde (92% de precisão e 89% de F1-score) e bons resultados gerais nas demais classes. Com uso de embeddings TF-IDF como entrada, o MLP provou ser uma solução confiável e menos sensível a variações em classes menores, como Meio Ambiente.





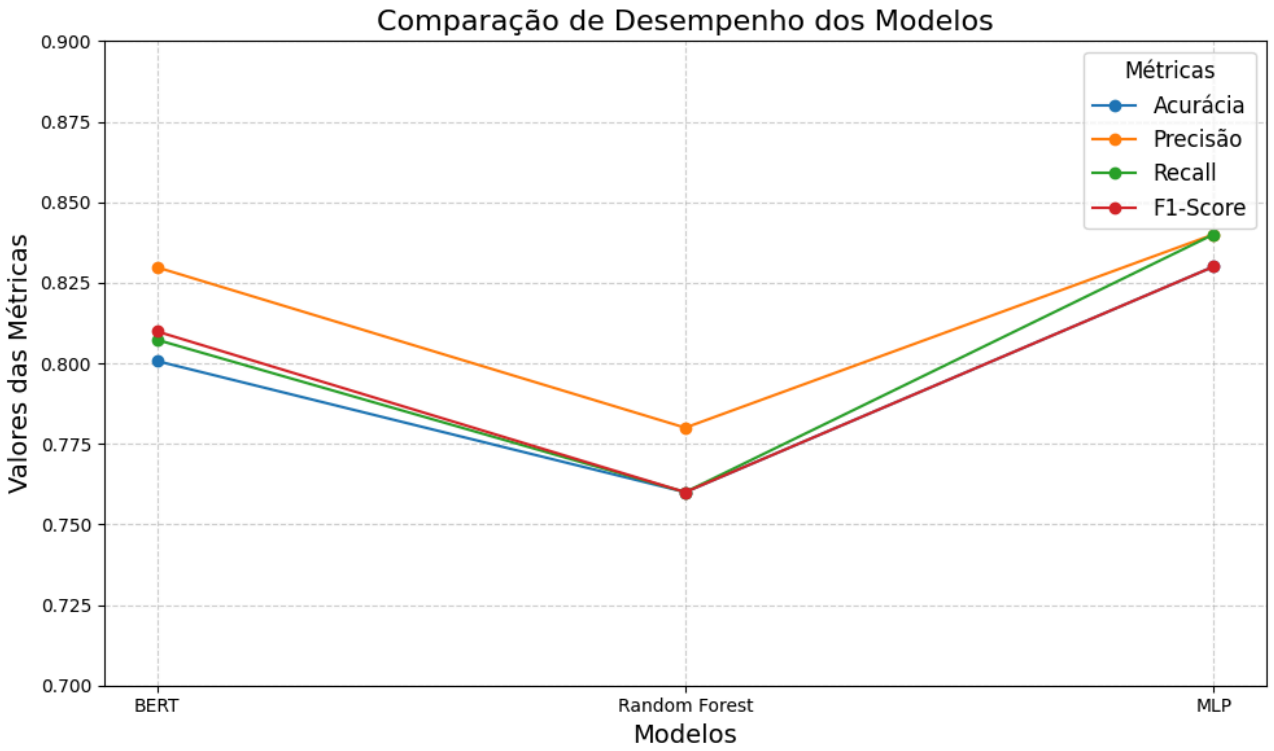


5.2 Visualização dos Resultados

As matrizes de confusão geradas para cada modelo revelaram os padrões e os principais desafios de cada abordagem:

- O BERT demonstrou maior precisão na separação de classes, especialmente em categorias como Saúde e Segurança, com erros menores nos eixos Meio Ambiente e Infraestrutura.
- O Random Forest foi eficaz em classes maiores, como Educação, mas mostrou confusões em categorias menos frequentes, como Meio Ambiente.
- O MLP, apesar de apresentar métricas gerais elevadas, teve dificuldade em capturar nuances contextuais entre classes semanticamente próximas.

Os gráficos de importância das palavras do Random Forest indicaram que termos como "escola", "saúde", "atendimento" e "meio ambiente" foram determinantes na classificação. Esses insights reforçam a relevância de representações vetoriais, como o TF-IDF, em modelos tradicionais.



5.3 Discussão dos Resultados

Com base nos resultados, observa-se que o BERT, mesmo com maior custo computacional, apresentou um desempenho mais equilibrado em termos de captura de padrões semânticos. Seu potencial para lidar com tarefas complexas torna-o uma opção robusta para futuros projetos de classificação de textos. Por outro lado, o MLP se destacou como uma solução prática e eficiente, alcançando métricas competitivas em menor tempo de treinamento. O Random Forest, por sua

vez, mostrou-se eficaz para aplicações que exigem simplicidade e boa interpretabilidade, embora tenha enfrentado dificuldades com dados de classes menos representadas.

De modo geral, os três modelos foram bem-sucedidos na tarefa de classificação multiclasse, com o MLP e o BERT destacando-se pelo melhor equilíbrio entre desempenho e robustez.

6. Conclusão

Neste projeto, realizamos a classificação automatizada de propostas governamentais do estado do Piauí, usando três abordagens distintas de aprendizado de máquina: Random Forest, MLP e BERT. O desempenho dos modelos foi avaliado em termos de acurácia, F1-score, precisão e recall, com o MLP apresentando o melhor desempenho geral, seguido pelo BERT e Random Forest.

A utilização de modelos baseados em transformers, como o BERT, provou ser uma abordagem eficaz para a tarefa, especialmente quando a complexidade semântica e os padrões contextuais são fatores importantes para a distinção entre categorias. O modelo MLP, apesar de ser menos complexo, se destacou em termos de desempenho em várias categorias, como Saúde, devido à sua maior estabilidade na tarefa de classificação multiclasse. Já o Random Forest, embora eficiente, teve alguns desafios ao lidar com classes menos frequentes e correlações semânticas mais sutis entre eixos.

O uso de técnicas clássicas de PLN, como a representação TF-IDF dos textos, também desempenhou um papel crucial no sucesso do Random Forest e do MLP, oferecendo uma maneira eficaz de transformar textos em representações numéricas úteis para os modelos.

Além disso, a avaliação dos modelos através de métricas como a matriz de confusão e o gráfico de importância das características forneceu insights valiosos sobre o comportamento de cada modelo e indicou áreas que podem ser exploradas em trabalhos futuros, como o balanceamento de classes e o uso de técnicas de ensemble para melhorar o desempenho e a robustez dos resultados.

Este projeto contribui para a área de Processamento de Linguagem Natural aplicado à análise política e representa um passo importante para o desenvolvimento de ferramentas que podem auxiliar na classificação automatizada de documentos governamentais, promovendo maior transparência e entendimento público sobre as propostas políticas.

O desempenho geral dos modelos indicou que é possível automatizar a classificação de propostas governamentais de forma eficaz, trazendo eficiência e escalabilidade para essa tarefa. Trabalhos futuros poderão investigar novas estratégias para otimizar o desempenho em conjuntos de dados desequilibrados e melhorar a precisão em categorias com características semânticas complexas.