UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: José Mário Oliveira Patrício

1. Introdução

O dataset tem como objetivo contribuir para o avanço dos sistemas de AES (Avaliação automática de redações). Em particular, o foco está na avaliação de redações narrativas escritas em português por estudantes do sistema de educação básica brasileiro. Dessa forma, os participantes são convidados a desenvolver um sistema computacional capaz de estimar uma nota para uma redação de entrada para cada competência especificada de interesse, seguindo a rubrica de avaliação estabelecida.

O conjunto de dados contém 1.235 redações escritas por estudantes do 5º ao 9º ano de escolas públicas do Brasil. Os estudantes foram instruídos a escrever uma redação narrativa baseada em um texto motivador. Todas as redações foram digitalizadas e anonimizadas manualmente. Em seguida, as redações foram analisadas por dois avaliadores humanos que avaliaram diferentes aspectos das redações com base em uma rubrica de correção pré-definida. Essa rubrica fornece orientações para que os educadores considerem quatro competências obrigatórias: (i) Registro Formal, (ii) Coerência Temática, (iii) Estrutura Retórica Narrativa e (iv) Coesão. Cada dimensão foi avaliada usando níveis inteiros que variam de 1 a 5, sendo que os níveis mais altos indicam melhor qualidade do texto e proficiência linguística, enquanto os níveis mais baixos demonstram falta de proficiência.

Pontos a se analisarem no dataset são uso de pontuações, palavras em maiúsculas, palavras escritas corretamente, símbolos específicos do dataset (ex, [P] representa parágrafo).

2. Observações

Como na descrição da atividade fala 'Neste exercício você deve utilizar um único classificador para aplicar no seu dataset, de acordo com a label escolhida', foi escolhida apenas a label 'cohesion', focando em identificar se o aluno consegue amarrar suas ideias dentro do seu texto. Além disso, o modelo de ML escolhido foi o Logistic Regression, por ser um classificador simples, rápido e funciona bem com TFIDF/BoW.

3. Resultados e discussão

Nesta seção deve-se descrever como foram as resoluções de cada questão. Crie sessões indicando a questão e discuta a implementação e resultados obtidos nesta. Explique o fluxograma do processo de cada questão, indicando quais processamentos são realizados nos dados. Sempre que possível, faça gráficos, mostre imagens, diagramas de blocos para que sua solução seja a mais completa possível. Discuta sempre sobre os números obtidos em busca de motivos de erros e acerto.

3.1 Item A)

No item a) foram utilizadas as seguintes técnicas de pré-processamento: remover_simbolos, remover_stopwords, remover_uppercase, remover_pontuacao, normalizar_espacos, stemming, lemmatization e um pipeline contendo todas as anteriores. Com isso, obteve-se os seguintes resultados:

Experimento Item A:

	Preprocessamento	Acurácia
0	Sem preprocessamento	0.651351
1	Remover maiúsculas	0.651351
2	Remover símbolos	0.648649
3	Remover stopwords	0.670270
4	Remover pontuação	0.643243
5	Stemming	0.637838
6	Lemmatização	0.629730
7	Normalizar espaços	0.651351
8	Pipeline completa	0.651351

Ou seja, todos os procedimentos retornaram resultados semelhantes, porém com a remoção dos stopwords a acurácia foi levemente superior. Vale ressaltar que nesse experimento foi utilizado o CountVectorizer como extrator de características. Como podemos perceber, a remoção de letras maiúsculas e a normalização de espaços obteve o mesmo resultado que sem processamento, ou seja, aparentemente esses pré-processamentos são inúteis para esse caso de prever a coesão textual. Remover pontuação e remover símbolos resulta em uma leve perda de acurácia, entrando no caso anterior. O stemming e a lemmatização resultaram em uma grande perda de acurácia, indicando que, nesse caso, é importante termos a palavra da maneira que ela foi escrita originalmente. Já a pipeline completa também foi idêntica ao sem processamento, ou seja, a combinação das técnicas utilizadas não gera um resultado melhor que cada uma separadamente. Por fim, a remoção de stopwords foi a única que

gerou um ganho de acurácia, indicando que a presença de palavras como "de", "da", entre outros, atrapalham na previsão da nota de coesão. Os resultados obtidos estão dentro do esperado, haja vista que a coesão textual depende mais da semântica do que da sintaxe do texto, ou seja, aplicar transformações nas palavras, formatar para letras minúsculas dificilmente irão ajudar um modelo a identificar melhor a nota da coesão textual.

3.1 Item B)

No item b), os seguintes extratores foram testados: ExtratorManual, CountVectorizer, TFIDF e Embeddings (modelo nilc-nlp/word2vec-cbow-50d). O ExtratorManual extrai as seguintes características: número de caracteres, número de palavras, número de frases, número de pontuações, número de dígitos, número de quebra de linhas e a contagem de palavras corretas. Desse modo, os seguintes resultados foram obtidos:

```
Experimento Item B:
                          Método  Acurácia
0      Manual COM preprocessing  0.713514
1         BoW COM preprocessing  0.670270
2      TF-IDF COM preprocessing  0.697297
3  Embeddings COM preprocessing  0.708108
```

Como podemos perceber, o extrator manual obteve o melhor resultado, em seguida temos a extração com Embeddings, depois TFIDF e por último BoW. Uma hipótese que pode explicar esse resultado é que o ExtratorManual utiliza features específicas que tendem a ter relação com a coesão textual, como por exemplo o número de pontuações, pois um texto com uma pontuação concisa pode ser mais fácil de ser entendido por um leitor.

3.1 Item C)

No item c) foi feita a comparação entre o Stemming e Lemmatization, utilizando o melhor extrator obtido no item c), ou seja, o ExtratorManual.

```
Experimento Item C:
                          Método  Acurácia
0      Stemming + ExtratorManual  0.700000
1  Lemmatização + ExtratorManual  0.713514
```

Nesse caso, a Lemmatização obteve um resultado levemente superior em relação ao Stemming, possivelmente porque o stemming pode cortar palavras de maneira que há uma perda na semântica que seria útil para identificar coesão.

4. Conclusões

Os resultados esperados foram satisfeitos? Se não, qual o motivo? Qual a sua análise?

Tendo em vista a natureza da tarefa, os resultados atuais (71% de acurácia) não são o esperado, haja vista a importância que a coesão textual tem em uma redação. Ou seja, alunos podem receber uma nota de coesão que não condiz com o texto escrito, prejudicando a percepção do professor sobre a redação do aluno. As hipóteses que podem explicar esse resultado são:

- *Limitação do modelo de machine learning, pois nesse estudo foi testado apenas a Regressão Logística.*
- *Falta de otimização dos hiperparâmetros, tanto dos extratores quanto do modelo de ML.*
- *Limitação dos dados,*

5. Próximos passos

- *Verificar mais formas de combinação de pré-processamento, por exemplo, testar apenas com remoção de stopwords, ou apenas com remoção de letras maiúsculas, a combinação das duas, o mesmo com a remoção de símbolos e etc.*
- *Adicionar mais atributos via ExtratorManual, como por exemplo a contagem de parágrafos (ex, símbolo [P]) pode ser um bom indicador da coesão do texto.*
- *Analisar quais features foram mais importantes para o ExtratorManual, verificando os coeficientes da Regressão Logística por exemplo.*
- *Combinar os extratores, por exemplo, utilizar ambas as features do TF-IDF junto com as do CountVectorizer, ou juntar TF-IDF com as features do ExtratorManual.*
- *Testar outras formas de classificação, como por exemplo a utilização de LLMs.*
- *Utilizar as outras variáveis do dataset como preditoras, como por exemplo a Estrutura Retórica Narrativa.*
- *Após obter a melhor opção de classificação, será possível criar um sistema web que automatiza a correção de redações, por exemplo, teria um campo onde o usuário irá fazer o upload de uma redação e automaticamente receberá sua nota.*