Como parte do processo de seleção, cada candidato deverá construir uma aplicação Web para resolver um exemplo de problema que poderia acontecer durante seu trabalho no projeto que você está sendo contratado. Neste projeto, trabalhamos com soluções computacionais para auxiliar a equipe do CAEd a avaliar o aprendizado de língua portuguesa. Dentre as soluções que criamos, estão soluções para processamento de áudio de leituras, criação de jogos, atribuição automática de notas para a capacidade leitora do aluno.
Apresentaremos para você abaixo um cenário hipotético de aplicação que poderia facilmente ser uma demanda para a nossa equipe de desenvolvimento.

1. Descrição do problema

Uma forma de avaliar a capacidade de leitura de uma criança é verificar a capacidade de decodificação, ou seja, a capacidade de uma criança ler sequências de palavras. Para isso, costuma-se utilizar testes com pequenos textos narrativos e com diferentes níveis de dificuldade. Vários fatores determinam o nível de dificuldade de um texto, como o seu tamanho (número de palavras), a familiaridade do leitor com as palavras que estão no texto, a estrutura gramatical (presença de apostos, etc), entre outros. O processo de criação de narrativas curtas é feito manualmente, com especialistas em avaliação de fluência. Contudo, conforme a avaliação tem se expandido para mais Estados do Brasil, há uma necessidade de automatizar a criação desses textos, uma vez usado o texto em uma avaliação, ele não pode ser utilizado novamente no futuro. Para auxiliar a equipe de criação de textos do CAEd, precisamos criar um sistema que gere novos textos. Abaixo está o passo-a-passo para resolver esse problema:

1. *[obrigatório]* Crie um modelo de predição que recebe um pequeno texto e classifique o texto de acordo com a sua complexidade. Você usará 4 níveis de complexidade: Ensino Fundamental I, Ensino Fundamental II, Ensino Médio e Ensino Superior. Essa tarefa será detalhada na seção 2 deste documento.
2. *[diferencial]* Crie um algoritmo que recebe um texto pequeno e gere um novo texto com inversões e troca de voz ativa/passiva. Verifique a complexidade do novo texto de acordo com o modelo de predição (veja seção 3 deste documento para dicas). Exemplo de alterações:
    a. Há duas semanas atrás estava chovendo => Estava chovendo há duas semanas atrás;
    b. O cantor Roberto Carlos fez aniversário => Roberto Carlos, o cantor, fez aniversário;
    c. O texto foi lido pelo aluno => O aluno leu o texto;

2. Criação do modelo de predição

- Para realizar essa tarefa, você receberá um dataset com textos já classificados nos 4 níveis de complexidade. Os textos representam trechos de conteúdo de materiais didáticos disponibilizados pelo MEC para os respectivos anos escolares. Cada arquivo do dataset representa trechos de um mesmo material didático.
- Faça uma análise dos dados fornecidos e teste mais de um modelo de predição para que consiga escolher adequadamente o que lhe entrega melhores resultados.
- Você também está livre para pré-processar o dataset (quebrar os arquivos em novas instâncias, por exemplos) e estruturar seus train/eval/test datasets da forma que desejar.
- Envie junto com o código e dados do projeto, um relatório contendo a sua metodologia para criação dos modelos, avaliação, etc. Analise adequadamente os resultados encontrados. A metodologia adotada para trabalhar os dados e analisar os resultados será principal critério de pontuação do seu projeto.

3. Verificando inversões em frases

- Existem diferentes estratégias que podem ser utilizadas para inverter frases. Você pode usar alguma biblioteca de processamento de linguagem natural (como o spacy) para gerar árvores de dependências e realizar operações na árvore para inverter a posição de alguns nós. Pode-se também criar uma solução mais simples baseada em regras sintáticas que você escolher. O importante aqui é documentar o seu processo de raciocínio para concluir a tarefa. Estamos mais interessados no processo do que no resultado final.
- Embora existam várias estratégias, sugere-se você tentar usar uma LLM (Large Language Model) caso você conheça essa solução, pois pode-se ter resultados mais rápidos.

- Independente da estratégia adotada, espera-se que você tenha um código que gere uma quantidade de novas sentenças para cada sentença de entrada e que as classifique utilizando sua solução do exercício anterior.

4. Instruções adicionais

- Utilize as ferramentas, linguagens de programação, frameworks, etc, que desejar.
- Caso implemente em python, existem algumas bibliotecas para manipulação de texto que podem te auxiliar, como o NLTK [1] e o spacy [2] (veja as árvores de *parsing*). Mas é importante frisar que seu uso não é obrigatório, você está livre para criar suas próprias soluções ou colocar as dependências que desejar no projeto.
- Monte as suas heurísticas para criar as funções de alteração do texto. Existem técnicas muito avançadas para isso, mas o mais importante é você ser criativo e projetar heurísticas interessantes, mesmo que simples.
- Se você não conseguiu completar todas as tarefas **não tem problema!** Mesmo não terminando tudo, envie ainda assim o que você fez, relate no relatório seus problemas. Nosso objetivo é avaliar suas estratégias. Queremos discutir contigo as ideias que teve. Mostre até onde chegou e como chegou. Adoramos discutir algoritmos!

5. Entrega

Os candidatos deverão enviar o código fonte, instruções para executar o sistema e um relatório informando como foi seu processo de desenvolvimento, ferramentas que usou, as ideias que teve para resolver cada um dos passos do projeto, experimentos com o modelo de predição, etc. Tudo deve ser encaminhado por e-mail para processoseletivo@fundacaocaed.org.br **até as 09:00 horas do dia 04/12/2023**.

Referências

[1] https://www.nltk.org/
[2] https://www.nltk.org/howto/wordnet.html