Projeto: HLD - Haskell Language Detector

Pedro de Luca Occulate Serra, RA: 21044215

1

Neste projeto, foi proposta a implementação de um algoritmo para detectar em qual idioma um texto está escrito, utilizando a linguagem *Haskell*. Para isso, foram utilizadas as traduções em inglês, francês, alemão, português e espanhol da *Declaração Universal dos Direitos Humanos* como textos de referência. A escolha deste texto se deu por duas razões: primeiro, por ser um mesmo texto traduzido em várias línguas, o que facilita a expansão para diferentes idiomas sem a necessidade de utilizar bases de dados distintas, o que poderia introduzir inconsistências no algoritmo. Além disso, a outra vantagem é que, sendo um texto de caráter político, ele utiliza uma ampla variedade de palavras, o que contribui para a eficácia do algoritmo.

Para alcançar o objetivo, utilizamos o conceito de N-gramas de palavras. Um N-grama de uma palavra é simplesmente uma sequência de n símbolos adjacentes dessa palavra, em ordem. Por exemplo, um *bi-grama* de "chocolate"seria a sequência ["ch", "ho", "oc", "co", "ol", "la", "at", "te"], enquanto o seu *tri-grama* seria ["cho", "oco", "ola", "ate"]. Esta técnica de **Processamento de Linguagem Natural (NLP)** compara a frequência de cada N-grama presente no **perfil da linguagem (Language Profile)** com aqueles presentes no texto cujo idioma se deseja determinar. O perfil da linguagem é uma sequência de N-gramas pré-processada da língua usada como referência na comparação. Esta técnica oferece algumas vantagens, como *simplicidade e eficiência*, pois é implementada utilizando métodos estatísticos simples que não requerem modelos complexos de *Machine Learning*, e *robustez a ruídos*, já que separa as palavras em pequenos *tokens*, desconsiderando pontuações, números, espaços, etc.

O funcionamento do algoritmo é simples: geram-se os perfis das línguas que se deseja comparar e o perfil do texto em questão. Em seguida, selecionam-se os primeiros K N-gramas (neste projeto foi utilizado o tri-grama) dos perfis e do texto, comparando-se a posição relativa de cada N-grama (caso o N-grama não seja encontrado, aplica-se uma penalização) e somam-se os valores dessas diferenças. Ao final, o perfil que gerar a soma de valor menor é o que corresponde à língua do texto.

A principal dificuldade na implementação deste algoritmo foi entender como utilizar a linguagem *Haskell* para realizar as operações necessárias, especialmente na geração da lista de frequência de N-gramas de cada língua. Além disso, compreender o funcionamento prático da própria linguagem já se mostrou um desafio. No entanto, à medida que o desenvolvimento progredia, pude perceber como certas operações se tornam de fato mais simples ao utilizar *Haskell* (como, por exemplo, aplicar uma função a diferentes valores de listas, manipular listas de diferentes formatos, remover caracteres indesejados dos textos, dentre outras). Além disso, a linguagem permite soluções muito diferentes para alguns problemas, o que desenvolve uma maneira de pensar mais focada em "o que são as coisas"e não apenas em "como essas coisas devem funcionar". Acredito que os pontos que mais gostei na minha implementação foi a maneira que gerei as listas de frequência, encadeando operações simples para solucionar algo complexo, e como o código ficou simples e facilmente navegável entre as funções.