Resumo 12 – PLN

Na décima segunda aula de Processamento de Linguagem Natural, o professor abordou as Tabelas de dispersão/espalhamento que, por sua vez, possui um universo U de chaves e um espaço amostral com “k” chaves. Uma dificuldade dessa abordagem é a possibilidade de duas chaves, “k1” e “k2” por exemplo, serem mapeadas para a mesma posição na tabela. Para evitar essas colisões, é possível aplicar uma função Hash que apresente um comportamento randômico, sendo esta uma função com o menor custo computacional possível . Vejamos algumas dessas funções.

O método da divisão recebe um número k e retorna o resto da divisão entre esse número e um outro número m. Os resultados estarão em um espaço amostral que vai de 0 até m – 1, mostrando assim a quantidade de elementos da tabela de dispersão. Em um exemplo passado em aula, temos os parâmetros m=12 e k=100, fazendo com que a função retorne 100 mod 12 = 4. O módulo, aqui, indica a nova posição do elemento. Quais são os melhores valores para o parâmetro m? A resposta para essa pergunta é: números primos. A justificativa não é trivial e o professor indicou que pesquisássemos em fóruns como o stack overflow. Um outro exemplo prático mostrou o tamanho ótimo para a tabela para n=200 elementos de 8 bits com, no máximo, 3 elementos colidindo.

O método da multiplicação utiliza uma constante A que aria entre 0 e 1. A vantagem deste método é que o valor de m não é crítico como no método da divisão. Para complementar, o professor passou alguns exemplos de valores para que possamos ter uma melhor noção sobre este método que, por sua vez, pode ser definido por: (kA mod 1) = kA – [kA].

Existem também outros métodos como o método da dobra, por exemplo. Entrando em detalhes sobre a função de hash, a grande vantagem é que trata-se de uma algoritmo simples e eficiente para inserção, remoção e busca. Dado que precisamos de uma função computacionalmente favorável, temos detalhes sobre o Feature Hashing. Um exemplo com um conjunto de dados foi mostrado para exemplificar o feature hashing, indicando um conjunto de dados, representando um idioma, com incríveis 40 milhões de palavras únicas (contando palavras novas e com erros de escrita).

Para consolidar os conceitos, o professor Jesus mostrou scripts em python (.py) com códigos utilizando as bibliotecas NumPy e afins utilizando como base as obras de Machado de Assis (análises de similaridade).

Feature hashing está relacionada a qualquer estrutura que precise de diminuição de espaço, em qualquer âmbito. A função de hashing pode mapear a estrutura grande em uma estrutura reduzida para agilizar o processamento.