Laboratório Nº 3: Text Mining - Métricas e Início do Processamento do Conjunto de Treino

Extração Automática de Informação
2021/2022

Prof. Joaquim Filipe
Eng. Filipe Mariano

Objetivos

- Continuação da implementação de vários passos de pré-processamento de modo a limpar o conteúdo dos documentos
- Introdução ao conceito de Term Frequency (TF), Inverse Document Frequency (IDF) e Term Frequency-Inverse Document Frequency (TF-IDF).

1. Contagens

Uma das *features* mais básicas que podem ser extraídas é o número de palavras ou caracteres de um texto. Estas *features* básicas podem ser tidas em conta, visto que, na generalidade dos casos, os sentimentos expressos de uma forma negativa podem conter um menor número de palavras do que os positivos. Contudo, esta afirmação pode também depender do domínio de aplicação do problema estudado.

2. Term Frequency

O conceito de *Term Frequency* (TF) é simplesmente o rácio entre a contagem do termo presente num texto e o número total de termos do texto. Desta forma, pode ser generalizado da seguinte forma:

```
TF = (Número de vezes que o termo T ocorre num documento) / (Número de termos do documento)
```

3. Inverse Document Frequency

Por sua vez, o *Inverse Document Frequency* (IDF) corresponde a uma métrica para determinar o quão raro um termo é entre os diversos documentos. O valor calculado tem o efeito de destacar palavras que são distintas (contêm informação útil) em um determinado documento. Assim, o IDF de um termo raro é alto, enquanto o de um termo frequente é provavelmente baixo.

O cálculo do IDF é feito através da seguinte fórmula:

$IDF_{(t)} = \log (N / d_{(t)})$

Em que `t` representa o termo, `N` o número de documentos do corpus e `d` o número de documentos em que o termo `t` ocorre.

4. TF-IDF

O *Term Frequency-Inverse Document Frequency* (TF-IDF) é uma estatística que tem como objetivo refletir o quão importante um termo é num documento, relativamente ao *corpus*. Ou seja, um valor alto de TF-IDF é alcançado através de uma frequência elevada do termo num documento e uma frequência baixa do mesmo termo em toda a coleção de documentos.

O cálculo do TF-IDF é feito através da seguinte fórmula:

$TF\text{-}IDF_{(t)} = TF_{(t)} * IDF_{(t)}$

Em que `t` representa o termo, `TF` a frequência do termo no documento e o `IDF` a frequência inversa do documento para esse termo.

5. Exercícios

1. Proceder à contagem de palavras e caracteres existentes num texto. Crie um módulo na diretoria `preprocessing` chamado `counting.js` que deverá exportar duas funções: `words` e `characters`. Ambas as funções deverão receber um texto como argumento e devolver o número de palavras ou de caracteres.

2. No módulo `counting.js` criar uma função `numberOfOccurrences` que receberá um termo e um texto e irá devolver o número de ocorrências desse termo.

3. Exporte uma função `exists` no módulo `counting.js` para que verifique a existência de um termo num texto. A função deverá receber uma string com um termo (unigrama ou bigrama) e o texto, e verificar que esse texto contém o termo que se pretende procurar, devolvendo um valor de verdadeiro ou falso.

4. Ainda no módulo `counting.js` criar uma função `tf` (term frequency) que receberá um termo e um texto e irá devolver a frequência desse termo. Devem contar o número de palavras existente no termo (já calculado a partir do `numberOfOccurrences`), para se saber o número total de termos que o texto tem. Por exemplo se a análise for feita para bigramas, o número total de termos será todos os conjuntos de 2 palavras seguidas.

5. No módulo `counting.js` criar uma função `idf` que receberá um o valor `N` (nº de documentos) e o valor de $d_{(t)}$ e irá devolver o cálculo do *inverse document frequency* (ver secção 3).

6. No módulo `counting.js` criar uma função `tfidf` que receberá o valor de tf e o valor idf e calculará o *term frequency - inverse document frequency* (ver secção 4).

7. No módulo `train.js` criar uma função `process`, incorporando o módulo de pré-processamento anteriormente criado na diretoria `preprocessing` (`index.js`) , de modo a realizar o seguinte:
a. Chamar a função `getTrainingSet` para obter todos os textos que estão marcados para serem utilizados como conjunto de treino.
b. Dividir o processamento por classes, pelo que deverá processar todos os textos de cada classe, ou seja, primeiro os papers da classe `happy` e depois os textos da classe `not happy`.
c. Para cada texto deverá utilizar a função definida no laboratório anterior para aplicar todas as técnicas referidas de pré-processamento abordadas para `n=1` e `n=2`, ou seja, para sequências de apenas 1 palavra e sequências de 2 palavras. Relembrar que este pré-processamento deveria de:
i. Remover as *stopwords* existentes em cada texto.
ii. Remover pontuação, números e espaços a mais existentes em cada texto.
iii. Normalizar as palavras existentes de acordo com o algoritmo de *Porter Stemming*.
d. Com o resultado obtido da alínea c, em que se pressupõe que tenha o texto "limpo" e normalizado e os *tokens* do texto, imprimir na consola os unigramas e bigramas de palavras encontrados em cada texto, assim

como o texto original e o texto processado.
e. Gravar o conteúdo impresso em consola, também para um ficheiro `training-process.txt`.

Nota: Ao finalizar este laboratório, já deverá conseguir processar um texto individualmente através da página elaborada na aula anterior, com as diversas técnicas de pré-processamento estudadas. Assim como, já deverá iniciar o processamento dos textos existentes no conjunto de treino para cada uma das classes definidas para o domínio de aplicação.