Projecto 2 PRI

Paulo Jorge Almeida dos Anjos 87822
Débora Maria Figueira Abreu 89242
João Miguel Correia de Sousa Silva Santos 67011
Instituto Superior Técnico, Lisboa, 07/12/2017

Exercício 1

Para este exercício implementou-se o algoritmo *PageRank* adaptado a frases de documentos. Implementou-se um grafo onde os nós correspondem às frases e as arestas referem-se aos pesos entre um par de frases.

Para testar este exercício os ficheiros devem ser colocados na pasta teste (alertamos para a existência de uma pasta de nome test, a pasta a ser utilizada neste exercício deverá ser a pasta teste e não test).

Exercício 2

O exercício 2 consistiu na implementação do algoritmo *PageRank*, com algumas alterações à solução apresentada no exercício 1. Implementou-se o algoritmo dado para este exercíco como apresentado no enunciado, onde a probabilidade inicial de cada frase é igual para todas as frases e o peso das arestas do grafo é *cosine similarity* entre duas frases(designou-se de solução base).

Partindo da solução dada no enunciado, para as probabilidades iniciais de cada frase, desenvolveu-se e testou-se as seguintes alternativas, numa tentativa de obter melhores resultados no cálculo de *mean average precision*:

1. Probabilidade inicial: $1/numerofrases$
2. Probabilidade inicial baseada na posição da frase no documento utilizando a seguinte fórmula $Prior(Pi) = P0 * (numerodefrases/i + 1)$, onde $P0 = 1/numerodefrases$ e i é a posição da frase Pi.
3. Probabilidade inicial baseada no score da frase contra o documento inteiro $Prior(Pi) = Po * score(i)$, onde $P0 = 1/numerodefrases$, i é a posição da frase Pi, e score(i) corresponde a *cosine similarity*
4. Probabilidade inicial baseada no número de termos da frase utilizando a seguinte fórmula $Prior(Pi) = numerodetermosdafrasei/numerotermosdodocumento$

   Para os pesos das arestas, para além da solução base em que os pesos das arestas correspondem a *cosine similarity* entre duas frases, implementou-se outra alternativa:

5. peso das arestas baseado no número de nounfrases em comum entre as duas frases da forma $Peso(i, j) = len(interception(nounfrases(i), nounfrases(j))$

MAP1: 0.07608322351667937
Para o cálculo da probabilidade utilizou-se (1) e para o cálculo dos pesos utilizou-se a solução base;

MAP2: 0.05717978444485795
Cálculo da Probabilidade: (2) e Cálculo dos pesos: solução base;

MAP3: 0.1809173846850317
Cálculo da Probabilidade: (3) e Cálculo dos pesos: solução base;

MAP4 : 0.1854990223420369
Cálculo da Probabilidade: (4) e Cálculo dos pesos: solução base;

MAP5: 0.06517290024790022
Cálculo da Probabilidade: (4) e Cálculo dos pesos: (5)

MAP6 : 0.042313915645900944
Cálculo da Probabilidade: (1) e Cálculo dos pesos: (5)

MAP7 : 0.04028803398398986
Cálculo da Probabilidade: (2) e Cálculo dos pesos: (5)

MAP8 : 0.10971605418002475
Cálculo da Probabilidade: (3) e Cálculo dos pesos: (5)

O melhor resultado obtido para *mean average precision* foi o MAP4 = 0.1809173846850317 onde utilizou-se para o cálculo da probabilidade inicial baseada no número de termos da frase e para o cálculo dos pesos a solução base.

Exercício 3

Para este exercício utilizou-se o algoritmo Perceptron da biblioteca sklearn.linear_model. O Perceptron foi treinado com o dataset indicado no enunciado TMario2006. As features utilizadas para descrever os documentos foram:

1. feature 1: baseada na posição da frase no documento com a seguinte fórmula : $f(i,j) = numfrasesdodocj/(posicaodafraseinodocj + 1)$;
2. feature 2: é baseada no score da *cosine similarity* da frase em relação a todo o documento;
3. feature 3: é o número de termos da frase i do documento j / número de termos do documento j;

Foi atribuído o valor de 1 às frases do documento que pertence ao resumo e 0 ás restantes. O method StandardScaler foi ulizado para uniformizar as features, removendo a média e para escalar a variância da unidade. Para selecionar as frases foi usado o perceptron.decision_function que devolve a confiança da classificação para cada uma das instâncias. Selecionou-se as intâncias com a confiança mais alta. O resultado obtido para a totalidade do dataSet foi MAP:0.17129695. Comparando com o melhor resultado do exercício anterior verificou-se que os valores para o MAP são melhores no exercício 2.

Exercico 4

Utilizou-se urlopen para ler os ficheiros xml directamente das páginas de notícias. Para fazer queries aos ficheiros xml utizou-se xml.etree.ElementTree. Efetuaram-se queries ao xml de modo a obter os elementos correspondentes às "descriptions". Todos os elementos correspondentes a

"description" foram agregados num só documento, De seguida realizou-se o resumo desse documento baseado na *cosine similarity* de cada frase relativamente ao documento todo. O resumo e o html são depois escritos para o ficheiro ./html/gost-host-one-page-template/index.html. A pasta html deve estar na mesma pasta dos ficheiros python.