Universidade Federal de Pernambuco – UFPE
Centro de Informática
Ciência da Computação
IF672 – Algoritmos e estruturas de dados
Professor: Renato Vimieiro
Projeto de Implementação **Entrega: 26/06/16**

Introdução

O objetivo deste trabalho é praticar o desenvolvimento de programas usando as técnicas estudadas em sala de aula, em particular, o uso de dicionários.

Você deve gastar pelo menos uma hora lendo este documento para se certificar que entendeu completamente o que é exigido, de forma a não perder tempo com idas e vindas ao documento para entender o que deve ser feito, e também não entregar algo que você acredite estar correto mas que depois descobre que se enganou com alguma especificação. Leia atentamente o documento destacando partes importantes, anotando as dúvidas para saná-las o quanto antes.

Análise de sentimentos

Este projeto é baseado em uma questão apresentada numa competição de programação sobre análise de sentimentos e mineração de dados/aprendizado de máquinas (https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews). Ele é uma adaptação do trabalho inicialmente proposto pelos professores Eric Manley e Timothy Urness da Drake University – EUA. O trabalho original está disponível em `http://nifty.stanford.edu/2016/manley-urness-movie-review-sentiment/`.

Análise de sentimentos é um processo computacional para identificar e caracterizar opiniões expressas em trechos de textos. O interesse é, sobretudo, em determinar se o autor do texto tem uma atitude positiva, negativa ou neutra em relação a um tópico, produto, etc.

Mineração de dados e Aprendizado de Máquina são áreas da ciência da computação que estudam algoritmos capazes de 'aprender' a partir de dados. Esses algoritmos buscam construir modelos que, a partir dos exemplos de entrada, conseguem fazer previsões ou tomar decisões para dados recebidos posteriormente.

O conjunto de dados que vamos 'aprender' é formado por 8.529 comentários sobre filmes postados no site Rotten Tomatoes. Cada comentário foi manualmente avaliado e assinalado um valor de sentimento entre 0 e 4. Os rótulos associados a cada valor são os seguintes:

0	negativo
1	razoavelmente negativo
2	neutro
3	razoavelmente positivo
4	positivo

O objetivo deste trabalho é construir um algoritmo que aprenda os dados de treinamento e, em seguida, determine os sentimentos de diversos comentários novos (não apresentados durante o treinamento, mas cujo valor de sentimento é conhecido).

Seu programa deve ler um arquivo contendo um conjunto de dados, chamado de **conjunto de treinamento**, com diversos comentários e seus respectivos escores (valores) de sentimento. O arquivo contendo os dados possui a seguinte estrutura: cada linha contém um escore e um comentário. O primeiro valor da linha é um número entre 0 e 4, indicando o escore do comentário; a partir daí, todos os demais símbolos pertencem aos comentário. Os arquivos a serem usados como entrada estão disponíveis no site da disciplina.

O conjunto de treinamento deve ser processado de forma a se computar os escores médios associados às palavras nos comentários. Cada palavra deve ser armazenada por triplas $(palavra, freq, escore)$, onde *palavra* é uma palavra pertencente a algum comentário no conjunto de treinamento, *freq* é o número de ocorrências dessa palavra, e *escore* é o escore (valor) de sentimento atribuído à palavra. O **escore** de uma palavra é a média dos sentimentos associados aos comentários onde a palavra ocorre.

Computado os escores médios das palavras no conjunto de treinamento, o programa deverá inferir o sentimento de um segundo conjunto de comentários, chamado de **conjunto de teste**. O conjunto de teste possui o mesmo formato do conjunto de treinamento: cada linha contém um comentário; o primeiro símbolo é um número entre 0 e 4, indicando o sentimento do comentário, o restante da linha contém o texto do comentário. O sentimento de um comentário é a média dos escores das palavras no comentário. Caso uma palavra não tenha ocorrido no conjunto de treinamento, deve-se atribuir o escore 2 a ela.

Inferido o escore de um comentário no conjunto de teste, o programa deve avaliar o desempenho de suas predições. Em outras palavras, ele deve computar a taxa de erros cometidos pelo programa. O erro de uma predição é a diferença entre o valor inferido e o valor real do sentimento de um comentário. Uma forma comum de avaliar o erro global do programa é através da soma dos quadrados dos erros. Assim, seu programa deve computar e exibir a soma dos quadrados dos erros.

Sua tarefa é desenvolver um programa que processe o conjunto de treinamento, armazenando os dados (palavras, freq e escore) em um dicionário. O dicionário deve ser de sua autoria e deve obedecer às boas práticas de programação orientada por objetos. A estrutura de dados por trás do dicionário fica a seu critério, mas existem algumas bonificações por marcos específicos.

Resumo: como o programa funciona

O programa exige que o usuário forneça exatamente duas strings: uma com o caminho do arquivo com o conjunto de treinamento e a outra com o caminho do conjunto de teste.

O programa deve primeiro computar o escore médio das palavras a partir do conjunto de treinamento. Em seguida, deve inferir o valor de sentimento de cada comentário no conjunto de teste. Feito isso, ele deve computar e exibir a soma dos quadrados dos erros.

O programa também deverá imprimir algumas estatísticas. A saber:

- tempo de execução (global e discriminado por tarefa – leitura de dados, computação dos escores e predição) e uso de memória
- quais são as palavras mais negativas e positivas no conjunto de treinamento
- quais são as palavras mais frequentes no conjunto de treinamento
- se implementar uma tabela hash, qual o número médio de colisões

Algumas definições: o que é uma palavra?

Antes de seguir, será útil definir o que é considerado uma palavra neste projeto. Definimos inicialmente o conceito de token. Um token é uma sequência de caracteres separados por espaços em branco. Uma palavra é definida como qualquer token não vazio que não é exclusivamente formado por pontuação. Você deverá encontrar as 'palavras' em um arquivo usando uma função *split_on_separators* para encontrar os tokens e uma função *clean_up* (a ser implementada por você) que serve para remover os símbolos de pontuação dos tokens. Se o resultado da função *clean_up* for uma string vazia, então ela não é considerada uma palavra. Para melhorar a eficiência do algoritmo, *clean_up* deve converter as letras para minúsculas. Dessa forma, após terem sido 'limpas', as palavras 'sim', 'Sim' e 'SIM!' serão todas iguais.

Logística

- O trabalho poderá ser feito em grupos de no máximo três componentes
- Os grupos devem se inscrever por **email**, enviando para o professor o nome e matrícula dos integrantes até o dia **10/06**.
- O código completo do programa deverá ser entregue por **email** até às **23h59 do dia 26/06/16**. Os códigos não entregues no horário serão penalizados com a perda de pontos.

- Trabalhos copiados de colegas ou da internet, seja trechos ou totalidade, serão prontamente anulados (todos os envolvidos).
- A apresentação dos projetos será no dia 29/06 impreterivelmente. Todos os membros devem saber todos os detalhes da implementação. O professor tem o direito de escolher a quem dirigirá a pergunta durante a apresentação.

Bônus

1. O código que for, pelo menos, 10% mais eficiente (rápido e/ou usar menos memória) que os concorrentes receberá bônus de 10% do total. Caso haja empate, ninguém receberá o bônus.