1. Definição do Problema.

A análise de sentimentos visa determinar automaticamente o sentimento presente em um texto. Neste trabalho entenderemos sentimento como polaridade, a saber negativa, neutra e positiva. Seu objetivo neste trabalho é utilizar os conhecimentos adquiridos na disciplina para tratar um problema de análise de sentimento em *tweets*.

A solução de análise de sentimentos será baseada em um *dicionário de sentimentos*. Um dicionário de sentimentos associa cada palavra a uma medida entre -1 (totalmente negativo) e 1 (totalmente positivo). Por exemplo, o termo excelente pode ter polaridade 0,9, o termo péssimo -0,85, e o termo hoje pode ter polaridade 0.

Dado que existe um dicionário de sentimentos, é possível calcular o sentimento do texto inteiro através da soma das polaridades individuais. Neste trabalho, assumiremos a seguinte escala:

- soma > 0,1: texto positivo
- soma < -0,1: texto negativo
- -0,1 >= soma <= 0,1: texto neutro

Por exemplo, dado os escores abaixo para três palavras contidas no dicionário, o tweet "Amo Garopa, maior astral !!" tem polaridade positiva. Note-se que nem todas palavras existem no dicionário (ex: Garopaba), e neste caso sua polaridade deve ser considerada como 0.

amo = 0,7; astral:0,3; maior:-0,1
Amo Garopapa, maior astral = 0,7+0+(-0,1)+0,3=0,9 > 0 = positivo

2. Problema

Neste trabalho você deve resolver dois problemas ligados à identificação de sentimento em tweets:

- Criar um *dicionário de sentimentos* a partir de um conjunto de tweets rotulados e usá-lo para determinar a polaridade de novos tweets;
- Implementar uma *segunda funcionalidade*, dentre as opções oferecidas no enunciado abaixo, as quais são uteis para entender o uso das palavras e sua relação com o sentimento.

Para resolver estes problemas, você deverá projetar e implementar sua solução utilizando duas estruturas de acesso a dados vistas na disciplina,. Os tweets devem ser armazenados em arquivo cujo formato e forma de acesso você vai escolher e justificar.

Sua aplicação deve poder:
a) ler tweets de uma entrada csv, armazená-los em um arquivo, e criar o dicionário e demais estruturas de acesso escolhidas;
b) adicionar mais tweets aos existentes a partir de um novo arquivo, mostrando que consegue atualizar o arquivo com os tweets armazenados, o dicionário e as demais estruturas de busca.
c) utilizar a funcionalidade de busca escolhida.

3. Dicionário de Sentimentos e Determinação de Polaridade de Tweets

a) Dicionário de Sentimentos

Para implementar seu dicionário, você deve escolher uma estrutura de dados adequada para representar a informação (conjunto de termos com informações relativas a polaridade), e fazer acesso a ela de forma eficiente.

Para criação/atualização do dicionário de sentimentos, você deve implementar o seguinte procedimento, dado um arquivo .csv que será fornecido.

Enquanto houverem tweets não lidos:

- Ler um tweet, o qual contém um texto e a polaridade do tweet (-1, 0 ou 1, representando negativo, neutro e positivo, respectivamente);
- extrair todas as palavras do tweet (desprezar pontuações, considerar apenas palavras com mais de 2 letras, converter tudo para letras minúsculas);
- Cada palavra do tweet recebe o escore do tweet;
- Atualizar o dicionário com cada palavra do tweet. Se a palavra:
    - não existe: incluir a palavra e associá-la com as seguintes informações <escore do sentimento, escore acumulado, número de tweets onde foi usada>;
    - já existe: atualizar as informações associadas, incluindo sentimento e contadores;
- gravar o tweet em um arquivo. Obs: este arquivo não é igual ao arquivo de entrado pois será indexado por outra estrutura de acesso, de acordo com a segunda funcionalidade escolhida.

Para exemplificar, considere que sua entrada tem os tweets abaixo:

-1	**Jogo** hoje, **maior** furada
1	Saida com a turma, **amo** muito !
1	**Amo** o Rio, **maior** astral
-1	**Jogo** foi a **maior** decepção
-1	Que decepção o **jogo**, **maior** roubada.
1	**Maior Jogo** hoje!
0	Saida do Rio hoje
0	Dia de **Jogo**

Ao final do procedimento, seu dicionário deve ter entradas para todas as palavras usadas. Quanto às informações associadas, considerando as palavras em negrito, o escore de sentimento, escore acumulado e número de tweets associados são:

jogo → <-0,4; -2; 5>
maior → <-0,2; -1; 5>
amo → <1, 2, 2>

Dicas:
- procure a função de tokenização de sua linguagem. Ela vai lhe ajudar a separar e selecionar as palavras;
- procure uma lista de "stop words" em português (o, a, os, as, etc), a fim de não criar entradas de pouco valor no dicionário. Seus resultados vão ficar melhores. Isto é opcional.
- coloque todas as entradas em letra minúsculas, para que Amo e amo não tenham escores distinto;
- funções avançadas como lematização/stemming podem ser usadas, mas não são o objeto do trabalho. A lematização busca a forma canônica e o stemming um radical. Por exemplo, *amarei* e *amarão* serão convertidos em *amar* na lematização, e em *am* com stemming.

Em resumo, você deve considerar que:
a) seu dicionário é uma representação dos termos que apareceram em um conjunto de tweets armazenados, e que vão auxiliar a definir a polaridade de novos tweets;
b) novos tweets sempre podem ser acrescentados ao conjunto existente, resultando na atualização tanto do dicionário, quanto do arquivo com o conjunto de tweets armazenados e suas estruturas de indexação correspondentes.

b) Determinação de Polaridade de Novos Tweets

Dado um dicionário construído a partir de um conjunto de tweets rotulados, você deve usá-lo para determinar a polaridade de novos tweets sem rótulo. Seu programa deve ler um arquivo com novos tweets SEM ROTULO, e gerar um arquivo que o associe com uma polaridade. Estes novo tweets **não** atualizam dicionário.

Considere o exemplo abaixo: um dicionário com as quatro palavras abaixo, e um arquivo CSV de antrada com o seguinte conteúdo :

jogo → <-0,4; -2; 5>
maior → <-0,2; -1; 5>
amo → <1, 2, 2>
demais → <0,8; 4;5>

NovosTweets.csv
Fui no jogo, demais !
Amo jogo.
Maior Jogo.

A saída deve ser um arquivo csv da seguinte forma (recomenda-se o uso de ; como separador, devido às vírgulas que podes estar no texto).

TweetsPolarizados.csv
Fui no jogo, demais !; 1
Amo jogo; 1
Maior decepção este jogo; -1

4. Funcionalidades adicionais

Para poder melhorar sua aplicação de análise de sentimentos, algumas funcionalidades adicionais permitirão entender o uso das palavras e sua relação com o sentimento. Você deve escolher **uma** dentre as funcionalidades abaixo, e projetar/implementar a funcionalidade utilizando uma estrutura de busca de apoio apropriada para o arquivo de tweets armazenado. Esta estrutura de acesso deve ser **diferente daquela utilizada para o dicionário**.

Lembre-se que cada modificação no conjunto de tweets armazenados deve ser refletida no dicionário e na estrutura de acesso escolhida para a segunda funcionalidade.

a) **Encontrar tweets com palavras específicas:** Dada uma palavra, gerar um arquivo csv com todos os tweets que a contém, e a respectiva polaridade. O critério de pesquisa para esta busca são: palavra, e opcionalmente a polaridade do tweet buscado. Usando os tweets do exemplo anterior, pode-se buscar todos os tweets que contenham a palavra *jogo* (os quatro tweets), ou somente os tweets com polaridade negativa que contenham a palavra *jogo* (somente dois tweets).

b) **Encontrar todas as variações de uma palavra baseado em um radical comum:** Dado um radical de duas ou mais letras, encontrar todas as suas variações encontradas no conjunto de tweets. Por exemplo, com o radical "am", poderiamos encontrar diversas conjugações do verbo amar (amei, amamos), variações do substantivo (amor, amores), formas de expressar sentimento (ameeeeeeeei, amooooo), além de outras palavras não relacionadas (amazonas).

c) **Encontrar palavras de acordo com o escore de sentimento**: Buscar palavras cujo escore sentimento esteja em algum ***intervalo*** (exemplo: palavras com escore maior que 0.3, ou escore entre -0.3..0.3

Você deve escolher **uma** das três funcionalidades acima, e justificar porque a estrutura de busca escolhida é adequada ao problema.

5. Definição dos Grupos

- O trabalho deve ser realizado em **duplas**.
- Para realizar em **triplas**, devem ser escolhidas duas funcionalidades de busca. Grupos de triplas historicamente demostram determinadas características, com raras exceções. Para aqueles que fizerem esta escolha, preparem-se para uma **avaliação oral rigorosa** quanto ao trabalho de cada membro grupo. Quando caronas forem detectados, todo grupo perderá nota.
- O trabalho individual deve ter uma boa razão, e ser autorizado pelo professor.

- A composição do grupo deve ser informada no moodle, via a tarefa definida para este fim.
- Todo o grupo deve estar presente na avaliação do trabalho. Caso algum colega esteja ausente, será interpretado que ele/ela não realizou o trabalho.
- Todos os membros do grupo devem ser capazes de responder sobre qualquer coisa do trabalho. **Ao responder no lugar de seu colega, você estará passando a mensagem que ele não trabalhou**.

6. O quê

Este trabalho exige a definição, projeto e implementação de uma solução aos problemas apresentados. Você deve produzir um relatório que explica detalhes importantes de seu projeto, justifica suas escolhas, e permitem compreender seu código. O código mostra que a solução foi implementada, e que está correta.

a) Material a ser entregue

- Relatório
- Código (.zip)

Tarefas serão abertas para enviar o material. **Não envie seu trabalho por email.**

b) Estrutura do Relatório

Seu relatório deve justificar suas escolhas, e explicar o projeto. O formato do relatório é livre. Ele deve conter:

a) A informação sobre os integrantes do grupo;
b) informação sobre a infraestrutura escolhida (ex: linguagem, framework, etc)
c) a organização e método de acesso do arquivo contendo os tweets, com justificativa da escolha;
d) a estrutura de dados para representar o dicionário, a justificativa da escolha, e os procedimentos de inserção e atualização;
e) a estrutura de dados para apoiar a funcionalidade de busca escolhida, a justificativa da escolha, e os procedimentos de inserção e atualização

7. Diretrizes

- O trabalho pode ser realizado em qualquer linguagem de programação que os alunos estimarem adequadas para o trabalho.
- Serão fornecidos arquivos de tweets, tanto para o desenvolvimento do trabalho, quanto para a demonstração quando da avaliação.
- **Não podem** ser utilizadas
    - **bibliotecas** com funções prontas de indexação ou determinação de polaridade, que podem ser apenas chamadas. **Seu código deve incluir a estrutura de pesquisa e suas respectivas operações ;**
    - **sistemas de gerência de dados** de qualquer natureza, apenas arquivos. Você deve utilizar as funções de acesso de arquivos disponíveis em sua linguagem

- Se você pegar código de terceiros, certifique-se que você entende de sua implementação. Você será questionado por suas escolhas, e se não demonstrar familiaridade com seu código, sua implementação **não** será considerada .
- **Casos de plágio serão tratados com a máxima severidade se detectados**.

8. Avaliação

- Relatório: 40% **(NAO NEGLIGENCIE O RELATÓRIO)**
- Implementação: 40%
- Questionamentos : 20% (pode ser individualizado)

9. Datas

- **Informe do Grupo** (via tarefa do moodle): 11/06
- **Entrega do material** (via tarefa do moodle): 01/07 – 20% de desconto sobre a nota total do trabalho com um dia dia de atraso, 10% por dia adicional de atraso
- **Demonstração e Avaliação Oral do Trabalho**: 03/07 e 05/07 (opcional 10/07 se necessário)

Este enunciado foi adaptado de uma idéia divulgada em http://nifty.stanford.edu/