

# ALGORITMIA AVANÇADA

## SPRINT D

TURMA 3NA

GRUPO 74

1140858	Carlos Filipe Borges Moutinho
1171602	Rui Manuel Castro Marinho
1181882	Rafael Martins Soares
1181892	Sara Santos Teixeira Silva
1181895	Fábio Alves da Silva

## CONTEÚDO

<b>Introdução .....</b>	<b>3</b>
<b>Desenvolvimento .....</b>	<b>4</b>
<b>Consideração de estados emocionais para encontrar caminhos .....</b>	<b>4</b>
A* .....	5
Best FirSt .....	7
Primeiro em profundidade.....	9
<b>Sugestão de Grupos .....</b>	<b>11</b>
<b>Cálculo de novos valores dos estados emocionais.....</b>	<b>13</b>
Efetuar esses cálculos para as Emoções Alegria e Angústia tendo como base a diferença entre Likes e Dislikes.....	13
Para as Emoções Esperança, Medo, Alívio e Deceção considere o facto de ser sugerido um ou mais utilizadores para um grupo do utilizador.....	15
Para as Emoções Orgulho, Remorso, Gratidão, Raiva considere o facto de um dado utilizador ter ou não ter um dado tag.....	17
<b>Estado da Arte do uso de metodologias/tecnologias (Língua Natural) aplicadas ao problema de tratamento de emoções em redes sociais de novos valores dos estados emocionais .....</b>	<b>20</b>
<b>Bibliografia .....</b>	<b>23</b>

# INTRODUÇÃO

O presente trabalho foi desenvolvido no âmbito da disciplina de Algoritmia Avançada (ALGAV), onde foi pedido o desenvolvimento do planeamento de contatos numa rede social. Para a mesma, foi utilizada a linguagem de programação PROLOG e utilizados métodos de pesquisa lecionados durante o semestre.

Neste relatório será efetuada o estudo das seguintes UC's:

- > Considerar estados emocionais para encontrar os caminhos.
- > Desenvolver um módulo que seja capaz de sugerir grupos, os quais terão de ter pelo menos N utilizadores e T tags comuns (N e T dados).
- > Calcular os novos valores dos estados emocionais em função de determinados objetivos.
- > Estudo do estado da arte de língua natural

# DESENVOLVIMENTO

## CONSIDERAÇÃO DE ESTADOS EMOCIONAIS PARA ENCONTRAR CAMINHOS

Para responder a este requisito incluímos a restrição do estado emocional na pesquisa de um caminho nos métodos A\*, BestFirst e Primeiro em profundidade implementados no Sprint C (considerando a força de ligação e relação) . Neste exercício, devem ser evitados qualquer utilizador um estado emocional negativo, isto é, com valores inferiores a 0,5.

A nossa base de conhecimento está organizada de forma a serem apresentados os estados emocionais dos utilizadores da seguinte forma:

```
mood('Sarah Silva',      0.5,0.5,0.5,0.5,0.5,0.5) .  
mood('Rafael Soares',    0.5,0.5,0.5,0.5,0.5,0.5) .  
mood('Dario Ornelas',    0.5,0.5,0.5,0.5,0.5,0.5) .  
mood('Fabio Silva',      0.5,0.5,0.5,0.5,0.5,0.5) .  
mood('Jonatas Granjeiro', 0.5,0.5,0.5,0.5,0.5,0.5) .  
mood('Pedro Mourao',     0.5,0.5,0.5,0.5,0.5,0.5) .  
mood('Carlos Moutinho',  0.5,0.5,0.5,0.5,0.5,0.5) .  
mood('Rui Mariske',      0.5,0.5,0.5,0.5,0.5,0.5) .
```

Cada campo representa respetivamente:

- Nome do user
- Estados emocionais Joy/distress
- Estados emocionais Hope/Fear
- Estados emocionais Relief/Disappointment
- Estados emocionais Pride/Remorse
- Estados emocionais Gratitude/Anger

Para cada emoção antagónica, o valor apresentado refere-se à emoção com notação positiva. Por exemplo, para as emoções Alívio e Deceção temos um valor de Alívio igual a 0,5 e um valor de Deceção igual a 0,5 (1-0,5).

```

aStar_Multicriterio_Mood(Orig, Dest, NivelMax, Cam, Custo) :-
    forcaMaximaMulticriterio_Mood(ListaOrdenada),
    aStar3_Mood(Dest, [(_ , 0, 0, [Orig])], NivelMax, 0, ListaOrdenada, Cam, Custo) .

aStar3_Mood(Dest, [(_ , Custo, _ , [Dest|T]) | _], _ , _ , _ , Cam, Custo) :-
    reverse([Dest|T], Cam) .

aStar3_Mood(Dest, [(_ , Ca, Ra, LA) | Outros], NivelMax, NivelAtual, ListaFMOrdenada, Cam, Custo) :-
    NivelAtual <= NivelMax,
    member(Act, LA),
    %LA=[Act|_],
    length(LA, NumUsers),
    NivelAtual1 is NumUsers-1,
    NivelAtual1 < NivelMax,
    nth0(NivelAtual1, ListaFMOrdenada, ValorEstimadoMax),

    findall((CEX, CaX, RaX, [X|LA]),
        (Dest \== Act,

        (ligacao(Act, X, ForcaX, RelacaoX), ligacao(X, Act, ForcaY, RelacaoY)),
        \+ member(X, LA),

        mood(X, Mood1, _ , _ , _ , _), Mood1 >= 0.5,
        mood(X, _ , Mood2, _ , _ , _), Mood2 >= 0.5,
        mood(X, _ , _ , Mood3, _ , _), Mood3 >= 0.5,
        mood(X, _ , _ , _ , Mood4, _), Mood4 >= 0.5,
        mood(X, _ , _ , _ , _ , Mood5), Mood5 >= 0.5,

        CaX is ForcaX + ForcaY + Ca,
        RaX is RelacaoX + RelacaoY + Ra,
        estimativa_multicriterio(ValorEstimadoMax, NivelMax, NivelAtual1, EstX),
        %write('Estimativa= '), write(EstX), nl,
        CEX is EstX),
        Novos),
    append(Outros, Novos, Todos),
    %write('Novos='), write(Novos), nl,
    sort(0, @>, Todos, TodosOrd),
    %write('TodosOrd='), write(TodosOrd), nl,
    aStar3_Mood(Dest, TodosOrd, NivelMax, NivelAtual1, ListaFMOrdenada, Cam, Custo) .

aStar3_Mood(Dest, [(_ , _ , _ , _ ) | Outros], NivelMax, NivelAtual, ListaOrdenada, Cam, Custo) :-
    aStar3_Mood(Dest, Outros, NivelMax, NivelAtual, ListaOrdenada, Cam, Custo) .

forcaMaximaMulticriterio_Mood(ListaFMOrdenada) :-
    findall(FactorMult,
        ((ligacao(Act, X, ForcaX, RelacaoX), ligacao(X, Act, ForcaY, RelacaoY)),
        Forca is ForcaX+ForcaY,
        Relacao is RelacaoX + RelacaoY,
        opcoes_multicriterio(Forca, Relacao, FactorMult)),
        ListaTotal),
    sort(0, @>=, ListaTotal, ListaFMOrdenada) .

```

Neste exercício ,no findall, quando é selecionado um próximo nó a juntar à lista, são previamente analisados os estados emocionais desse mesmo utilizador. Apenas são permitidos utilizadores cujo estado emocional seja positivo, isto é, superior ou igual a 0,5.

### EXEMPLO

Tendo por base a base de conhecimento abaixo mostramos o resultado obtido pelo algoritmo A\* multicritério desenvolvido no Sprint C e o A\* multicritério desenvolvido para responder a este requisito.

```
mood('Sarah Silva', 0.6,0.5,0.5,0.5,0.5,0.5).
mood('Rafael Soares', 0.5,0.4,0.4,0.5,0.5,0.5).
mood('Dario Ornelas', 0.5,0.5,0.5,0.5,0.5,0.5).
mood('Fabio Silva', 0.5,0.5,0.6,0.5,0.5,0.5).
mood('Jonatas Granjeiro', 0.5,0.4,0.5,0.5,0.5,0.5).
mood('Pedro Mourao', 0.5,0.5,0.5,0.5,0.5,0.5).
mood('Carlos Moutinho', 0.5,0.5,0.5,0.5,0.5,0.5).
mood('Rui Mariske', 0.5,0.4,0.5,0.5,0.5,0.5).
```

A\* multicritério simples:

```
[debug] ?- aStar_Multicriterio('Sarah Silva', 'Pedro Mourao',4,C,M).
C = ['Sarah Silva', 'Rafael Soares', 'Pedro Mourao'],
```

A\* multicritério considerando os estados emocionais

```
| aStar_Multicriterio_Mood('Sarah Silva', 'Pedro Mourao',4,C,M).
C = ['Sarah Silva', 'Dario Ornelas', 'Fabio Silva', 'Pedro Mourao'],
```

Como visto acima o resultado obtido através do A\* multicritério simples foi [Sarah Silva, Rafael Soares, Pedro Mourao]. Para testar a eficácia do nosso algoritmo decidimos aplicar ao utilizador Rafael Soares uma emoção negativa. Tal como era suposto, verificamos na imagem acima que, quando utilizado o algoritmo com os estados emocionais, o caminho resultante é outro, tendo sido descartado o utilizador Rafael Soares .

## BEST FIRST

```
bestfs1_Mult_Mood(Orig, Dest, NivelMax, Cam, Custo) :-
    bestfs12_Mult_Mood(Dest, [[Orig]], NivelMax, 0, Cam, Custo).
    %write('Caminho='), write(Cam), nl.

bestfs12_Mult_Mood(Dest, [[Dest|T]|_], _, _, Cam, Custo) :-
    reverse([Dest|T], Cam),
    calcula_custo(Cam, Custo).

bestfs12_Mult_Mood(Dest, [[Dest|_] | LLA2], NivelMax, NivelAtual, Cam, Custo) :-
    !,
    bestfs12_Mult_Mood(Dest, LLA2, NivelMax, NivelAtual, Cam, Custo).

bestfs12_Mult_Mood(Dest, LLA, NivelMax, NivelAtual, Cam, Custo) :-
    NivelAtual < NivelMax,
    LA=[Act|_],
    member1(LA, LLA, LLA1),

    length(LA, NumUsers),
    NivelAtual1 is NumUsers-1,

    ((Act==Dest, !, bestfs12_Mult_Mood(Dest, [LA|LLA1], NivelMax, NivelAtual, Cam, Custo));
    (NivelAtual1 < NivelMax,
    findall((FatorMult, [X|LA]),
        (ligacao(Act, X, FLX, FRX), ligacao(X, Act, FLY, FRY),

        mood(X, Mood1, _, _, _, _) , Mood1 >= 0.5,
        mood(X, _, Mood2, _, _, _) , Mood2 >= 0.5,
        mood(X, _, _, Mood3, _, _) , Mood3 >= 0.5,
        mood(X, _, _, _, Mood4, _) , Mood4 >= 0.5,
        mood(X, _, _, _, _, Mood5, _) , Mood5 >= 0.5,

        Ligacao is FLX + FLY,
        Relacao is FRX + FRY,
        opcoes_multicriterio(Ligacao, Relacao, FatorMult),
        \+member(X, LA)),
        Novos),
    Novos \= [], !,
    %nl, write('Novos= '), write(Novos), nl,
    sort(0, @>=, Novos, NovosOrd),
    %nl, write('Novos Ord= '), write(NovosOrd), nl,

    retira_custos(NovosOrd, NovosOrd1),
    append(NovosOrd1, LLA1, LLA2),
    %nl, write('LLA2= '), write(LLA2), nl,
    bestfs12_Mult_Mood(Dest, LLA2, NivelMax, NivelAtual1, Cam, Custo)).
```

Neste exercício ,no findall, quando é selecionado um próximo nó a juntar à lista, são previamente analisados os estados emocionais desse mesmo utilizador. Apenas são permitidos utilizadores cujo estado emocional seja positivo, isto é, superior ou igual a 0,5.

### EXEMPLO

Tendo por base a base de conhecimento abaixo mostramos o resultado obtido pelo algoritmo Best First\* multicritério desenvolvido no Sprint C e o Best First multicritério desenvolvido para responder a este requisito.

mood('Sarah Silva',	0.6,0.5,0.5,0.5,0.5,0.5).
mood('Rafael Soares',	0.5,0.6,0.7,0.5,0.5,0.5).
mood('Dario Ornelas',	0.5,0.5,0.4,0.5,0.5,0.5).
mood('Fabio Silva',	0.5,0.5,0.6,0.5,0.5,0.5).
mood('Jonatas Granjeiro',	0.5,0.5,0.5,0.5,0.5,0.5).
mood('Pedro Mourao',	0.5,0.5,0.5,0.5,0.5,0.5).
mood('Carlos Moutinho',	0.5,0.5,0.5,0.5,0.5,0.5).
mood('Rui Mariske',	0.5,0.4,0.5,0.5,0.5,0.5).

Best First multicritério simples:

```
| bestfs1_Mult('Sarah Silva', 'Pedro Mourao',5,C,L).  
C = ['Sarah Silva', 'Dario Ornelas', 'Fabio Silva', 'Pedro Mourao'],
```

Best First multicritério considerando os estados emocionais

```
[debug] ?- bestfs1_Mult_Mood('Sarah Silva', 'Pedro Mourao',5,C,Custo).  
C = ['Sarah Silva', 'Rafael Soares', 'Jonatas Granjeiro', 'Pedro Mourao'],
```

Como visto acima ,o resultado obtido através do Best First multicritério simples foi [Sarah Silva, Dario Ornelas, Fabio Silva, Pedro Mourao]. Para testar a eficácia do nosso algoritmo decidimos aplicar ao utilizador Dario Ornelas uma emoção negativa. Tal como era suposto, verificamos na imagem acima que, quando utilizado o algoritmo com os estados emocionais, o caminho resultante é outro, tendo sido descartado o utilizador Dario Ornelas .



## PRIMEIRO EM PROFUNDIDADE

```
caminho_maisForte2_Mult_Mood(Orig, Dest, Nivel, LCaminho_maisForte, Forca) :-  
    %get_time(Ti),  
    (melhor_caminho_maisForte2_Mult_Mood(Orig, Dest, Nivel); true),  
    retract(melhor_sol_maisForte_Mult_Mood(LCaminho_maisForte, Forca)).  
    %retract(conta_sol_maisForte_Mult(NS1)),  
    %get_time(Tf),  
    %T is Tf-Ti,  
    %nl, write('Nº:ºmero de solui:ºi:ºes encontradas: '), write(NS1), nl,  
    %nl, write('Tempo de geracao da solucao:'), write(T), nl.  
  
melhor_caminho_maisForte2_Mult_Mood(Orig, Dest, Nivel) :-  
    asserta(melhor_sol_maisForte_Mult_Mood(_, 0)),  
    asserta(conta_sol_maisForte_Mult_Mood(0)),  
    dfs(Orig, Dest, LCaminho),  
  
    checkMoods(LCaminho, [], List), !,  
    length(List, LCaminho1),  
    LCaminho1 > 0,  
  
    atualiza_melhor_maisForte2_Mult_Mood(LCaminho, Nivel),  
    fail.  
  
atualiza_melhor_maisForte2_Mult_Mood(LCaminho, Nivel) :-  
    retract(conta_sol_maisForte_Mult_Mood(NS)),  
    NS1 is NS+1,  
    asserta(conta_sol_maisForte_Mult_Mood(NS1)),  
    melhor_sol_maisForte_Mult_Mood(_, N),  
  
    calculateStrength2_Mult_Mood(LCaminho, [], SumStrength),  
    length(LCaminho, C),  
    %nl, write(LCaminho), nl,  
    C1 is C-1,  
    C1 <= Nivel, !,  
    SumStrength > N,  
    retract(melhor_sol_maisForte_Mult_Mood(_, _)),  
    asserta(melhor_sol_maisForte_Mult_Mood(LCaminho, SumStrength)).  
  
calculateStrength2_Mult_Mood([], Temp, SumStrength) :-  
    sumList(Temp, SumStrength), !.  
  
calculateStrength2_Mult_Mood([A, B|T], Temp, Strength) :-  
    ligacao(A, B, FL1, FR1),  
    ligacao(B, A, FL2, FR2),  
    Forca is FL1+FL2,  
    Relacao is FR1+FR2,  
    opcoes_multicriterio(Forca, Relacao, Soma),  
    removeElement([A, B|T], A, List),  
    calculateStrength2_Mult_Mood(List, [Soma|Temp], Strength).
```

```
checkMoods([], Temp, FinalList) :- reverse(Temp, FinalList).  
checkMoods([H|T], Temp, List) :-  
    mood(H, Mood1, Mood2, Mood3, Mood4, Mood5, _),  
    Mood1 >= 0.5,  
    Mood2 >= 0.5,  
    Mood3 >= 0.5,  
    Mood4 >= 0.5,  
    Mood5 >= 0.5,  
    checkMoods(T, [H|Temp], List).  
checkMoods(_, _, List) :- checkMoods([], [], List).
```

Neste exercício, uma vez chamado o DFS decidimos verificar os estados emocionais de cada utilizador apresentado no caminho resultante. Para tal implementamos o CheckMood que vai iterar todos os utilizadores do caminho e verificar se todos têm estados emocionais positivos. Caso não seja o caso, o caminho será descartado.

#### EXEMPLO

Tendo por base a base de conhecimento abaixo mostramos o resultado obtido pelo algoritmo Primeiro em profundidade multicritério desenvolvido no Sprint C e o Primeiro em profundidade multicritério desenvolvido para responder a este requisito.

```
mood('Sarah Silva', 0.6,0.5,0.5,0.5,0.5,0.5) .  
mood('Rafael Soares', 0.5,0.6,0.7,0.5,0.5,0.5) .  
mood('Dario Ornelas', 0.4,0.5,0.6,0.5,0.5,0.5) .  
mood('Fabio Silva', 0.5,0.5,0.6,0.5,0.5,0.5) .  
mood('Jonatas Granjeiro', 0.5,0.5,0.5,0.5,0.5,0.5) .  
mood('Pedro Mourao', 0.5,0.5,0.5,0.5,0.5,0.5) .  
mood('Carlos Moutinho', 0.5,0.5,0.5,0.5,0.5,0.5) .  
mood('Rui Mariske', 0.5,0.4,0.5,0.5,0.5,0.5) .
```

Primeiro em profundidade multicritério simples:

```
| caminho_maisForte2_Mult('Sarah Silva', 'Pedro Mourao',4,L,C).  
L = ['Sarah Silva', 'Dario Ornelas', 'Fabio Silva', 'Pedro Mourao'],
```

Primeiro em profundidade multicritério considerando os estados emocionais

```
[debug] ?- caminho_maisForte2_Mult_Mood('Sarah Silva', 'Pedro Mourao',4,L,C).  
L = ['Sarah Silva', 'Rafael Soares', 'Fabio Silva', 'Pedro Mourao'],
```

Como visto acima, o resultado obtido através do Primeiro em profundidade multicritério simples foi [Sarah Silva, Dario Ornelas, Fabio Silva, Pedro Mourao]. Para testar a eficácia do nosso algoritmo decidimos aplicar ao utilizador Dario Ornelas uma emoção negativa. Tal como era suposto, verificamos na imagem acima que, quando utilizado o algoritmo com os estados emocionais, o caminho resultante é outro, tendo sido descartado o utilizador Dario Ornelas .

## SUGESTÃO DE GRUPOS

Este requisito solicita a sugestão do maior grupo com base dos seguintes critérios:

- um mínimo de N utilizadores
- Número de tags em comum
- Lista de tags obrigatórias

### PROLOG

```
suggestGroups (User, NumCommonTags, MinNumberUsers, MandatoryListTags, Group) :-  
    length (MandatoryListTags, Length),  
    (Length > 0, !, suggestGroups_MandatoryTags (User, NumCommonTags, MinNumberUsers, MandatoryListTags, Group)) ;  
    suggestGroups_NoMandatoryTags (User, NumCommonTags, MinNumberUsers, Group) .  
  
suggestGroups_MandatoryTags (User, NumCommonTags, MinNumberUsers, MandatoryTags, Group) :-  
    userWithXTagsInCommon (NumCommonTags, ListAllCombinations),  
    filterByUserOrig (User, ListAllCombinations, [], FilteredByUser),  
    filter_RemoveUserOrig (User, FilteredByUser, FilteredByNonOrig),  
    filterByMandatoryTags (FilteredByNonOrig, MandatoryTags, FilteredByMandatoryTags),  
    filterByNumberOfUsers (MinNumberUsers, FilteredByMandatoryTags, [], FilteredByNumberOfUsers), !,  
    length (FilteredByNumberOfUsers, FilteredByNumberOfUsersLength),  
    FilteredByNumberOfUsersLength > 0,  
    select_BiggestGroup (FilteredByNumberOfUsers, [], Group) .  
  
suggestGroups_NoMandatoryTags (User, NumCommonTags, MinNumberUsers, Group) :-  
    userWithXTagsInCommon (NumCommonTags, ListAllCombinations),  
    filterByUserOrig (User, ListAllCombinations, [], FilteredByUser),  
    filter_RemoveUserOrig (User, FilteredByUser, FilteredByNonUser),  
    filterByNumberOfUsers (MinNumberUsers, FilteredByNonUser, [], FilteredByNumberOfUsers),  
    select_BiggestGroup (FilteredByNumberOfUsers, [], Group) .
```

### EXPLICAÇÃO

Neste algoritmo começamos por verificar se existem tags impostas e dependendo da sua existência será reencaminhado para `suggestGroups_MandatoryTags` ou para `suggestGroups_NoMandatoryTags`.

De seguida, obtemos a lista de todas as combinações de X tags usando o método desenvolvido no sprint anterior. Uma vez que temos todas as combinações possíveis, filtramos a lista a uma lista de combinações que sejam comuns ao `userOrig`. Esta manipulação permite filtrar as tags em comum com o utilizador.

Posteriormente, iteramos as combinações ainda disponíveis e apenas selecionamos aquelas que têm pelo menos um número mínimo de utilizadores imposto e no caso de haver tags mandatórias, será feita a validação da presença das tags. Uma vez toda as filtrações feitas, retornamos o maior dos grupos.

### EXEMPLOS

?- suggestGroups('Sarah Silva', 2,2,[],L).

L = ['Rui Mariske', 'Carlos Moutinho', 'Jonatas Granjeiro', 'Fabio Silva', 'Dario Ornelas'] .

?- suggestGroups('Sarah Silva', 2,3,[],L).

L = ['Rui Mariske', 'Carlos Moutinho', 'Jonatas Granjeiro', 'Fabio Silva', 'Dario Ornelas'] .

?- suggestGroups('Sarah Silva', 3,2,[],L).

L = ['Jonatas Granjeiro', 'Fabio Silva'] .

?- suggestGroups('Sarah Silva', 2,3,[travel],L).

L = ['Pedro Mourao', 'Fabio Silva', 'Dario Ornelas', 'Rafael Soares'].

?- suggestGroups('Sarah Silva', 2,2,[music, travel],L).

L = ['Fabio Silva', 'Dario Ornelas'].

?- suggestGroups('Sarah Silva', 20,3,[],L).

**false.**

## CÁLCULO DE NOVOS VALORES DOS ESTADOS EMOCIONAIS

EFETUAR ESSES CÁLCULOS PARA AS EMOÇÕES ALEGRIA E ANGÚSTIA TENDO COMO BASE A DIFERENÇA ENTRE LIKES E DISLIKES

```
moodUpdate_Joy_Distress(User, MoodUpdatedValue) :-
:
    getMood_Joy_Distress(User, UserMoodValue),
    getLikesDislikes(User, LikesDislikesValue),
    abs(LikesDislikesValue, LikesDislikesValueAbs),
    SaturationValue is 50,
    append([LikesDislikesValueAbs], [SaturationValue], List),
    getMin(List, MinValue),
    getMoodUpdate(UserMoodValue, LikesDislikesValue, SaturationValue, MinValue, MoodUpdatedValue).

getMoodUpdate(UserMoodValue, LikesDislikesValue, SaturationValue, MinValue, MoodUpdatedValue) :-
    (LikesDislikesValue > 0,
     MoodUpdatedValue is (UserMoodValue + (1 - UserMoodValue) * (MinValue / SaturationValue)));
    (LikesDislikesValue < 0,
     MoodUpdatedValue is (UserMoodValue * (1 - (MinValue / SaturationValue)))).

getMoodUpdate(UserMoodValue, _, _, _, UserMoodValue).

getMood_Joy_Distress(User, UserMoodValue) :-
    mood(User, UserMoodValue, _, _, _, _).

getLikesDislikes(User, Value) :-
    findall(FR, ligacao(_, User, _, FR), ListForcaRelacao),
    sumList(ListForcaRelacao, Value).

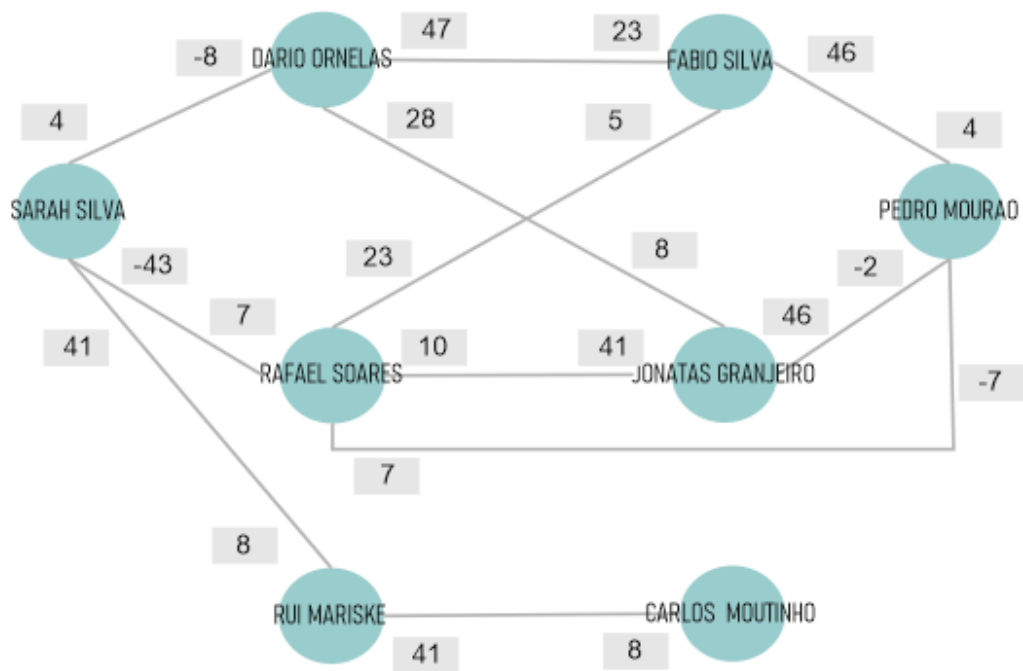
getMin(L, MinValue) :-
    sort(L, SortedList),
    nth0(0, SortedList, MinValue).
```

Iniciamos por obter o mood atual do utilizador para a Alegria/angustia e da diferença entre likes e dislikes. De seguida, determinamos qual o nosso valor de saturação e obtemos o valor mínimo entre este e o valor absoluto da diferença de likes e dislikes.

Se a diferença entre likes e dislikes for positiva então utilizamos a equação que leva ao aumento da emoção. Caso a diferença for negativa, utilizamos a equação que leva ao decréscimo da emoção. Se não houver diferenças entre os valores da emoção mantém-se.

### EXEMPLO

```
?- moodUpdate_Joy_Distress('Sarah Silva',L).  
L = 0.57000000000000001 .  
  
?- moodUpdate_Joy_Distress('Rafael Soares',L).  
L = 0.46 .
```



Se olharmos em detalhe para as forças de relação da Sarah e do Rafael e as somarmos temos:

- Sarah: **7**
- Rafael: **-4**

Perante estes resultados era expectável um aumento da emoção de Alegria na Sarah e um aumento da Angústia no Rafael (que se reflete por uma diminuição do valor).

Este é um exemplo que valida o nosso algoritmo.

PARA AS EMOÇÕES ESPERANÇA, MEDO, ALÍVIO E DECEÇÃO CONSIDERE O FACTO DE SER SUGERIDO UM OU MAIS UTILIZADORES PARA UM GRUPO DO UTILIZADOR.

```
moodUpdate_EM_AD(User, ListSuggestion, ListWantedUsers, ListUnwantedUsers, MoodUpdate_Esp, MoodUpdate_Med, MoodUpdate_Ali, MoodUpdate_Dec) :-
    get_UserMood_EspMed(User, UserMood_EspMed),
    get_UserMood_AliDec(User, UserMood_AliDec),
    intercecao(ListSuggestion, ListWantedUsers, ListWanted_Included), %Lista de users desejados incluidos
    intercecao(ListSuggestion, ListUnwantedUsers, ListUnwanted_Included), %Lista de users indesejados incluidos
    get_UsersExcluded(ListWanted_Included, ListWantedUsers, ListWanted_Excluded), %Lista de users desejados excluidos ( )
    get_UsersExcluded(ListUnwanted_Included, ListUnwantedUsers, ListUnwanted_Excluded), %Lista de users indesejados excluidos

    %Variaveis
    length(ListWantedUsers, A), Num_AllWantedUsers is A,
    length(ListUnwantedUsers, B), Num_AllUnwantedUsers is B,
    length(ListWanted_Included, C), Num_WantedUsers_Included is C, % Esperanca
    length(ListWanted_Excluded, D), Num_WantedUsers_Excluded is D, % Dececao
    length(ListUnwanted_Included, E), Num_UnwantedUsers_Included is E, % Medo
    length(ListUnwanted_Excluded, F), Num_UnwantedUsers_Excluded is F, % Alivio

    %Media
    Media_EspDec is (Num_AllWantedUsers/2),
    Media_AliMed is (Num_AllUnwantedUsers/2),

    %Ratios (V/Max.V)
    Esperanca_Ratio is (Num_WantedUsers_Included/Num_AllWantedUsers),
    Medo_Ratio is (Num_UnwantedUsers_Included/Num_AllUnwantedUsers),
    Alivio_Ratio is (Num_UnwantedUsers_Excluded/Num_AllUnwantedUsers),
    Dececao_Ratio is (Num_WantedUsers_Excluded/Num_AllWantedUsers),

    % Calculos Part I: Esperanca e Dececao
    % Esperanca
    ( (Num_WantedUsers_Included > Media_EspDec, moodUpdate_Up(UserMood_EspMed, Esperanca_Ratio, MoodUpdate_EspMed_1));
      (Num_WantedUsers_Included < Media_EspDec, moodUpdate_Down(UserMood_EspMed, Esperanca_Ratio, MoodUpdate_EspMed_1));
      (MoodUpdate_EspMed_1 is UserMood_EspMed)),
    % Dececao
    ( (Num_WantedUsers_Excluded > Media_EspDec, moodUpdate_Down(UserMood_AliDec, Dececao_Ratio, MoodUpdate_AliDec_1));
      (Num_WantedUsers_Excluded < Media_EspDec, moodUpdate_Up(UserMood_AliDec, Dececao_Ratio, MoodUpdate_AliDec_1));
      (MoodUpdate_AliDec_1 is UserMood_AliDec)),

    % Calculos Part II: Alivio e Medo
    % Alivio
    ( (Num_UnwantedUsers_Excluded > Media_AliMed, moodUpdate_Up(MoodUpdate_AliDec_1, Alivio_Ratio, MoodUpdate_AliDec));
      (Num_UnwantedUsers_Excluded < Media_AliMed, moodUpdate_Down(MoodUpdate_AliDec_1, Alivio_Ratio, MoodUpdate_AliDec));
      (MoodUpdate_AliDec is MoodUpdate_AliDec_1)),
    % Medo
    ( (Num_UnwantedUsers_Included > Media_AliMed, moodUpdate_Down(MoodUpdate_EspMed_1, Medo_Ratio, MoodUpdate_EspMed));
      (Num_UnwantedUsers_Included < Media_AliMed, moodUpdate_Up(MoodUpdate_EspMed_1, Medo_Ratio, MoodUpdate_EspMed));
      (MoodUpdate_EspMed is MoodUpdate_EspMed_1)),

    % var os valores por emocao
    MoodUpdate_Esp is (MoodUpdate_EspMed),
    MoodUpdate_Med is (1-MoodUpdate_EspMed),
    MoodUpdate_Ali is (MoodUpdate_AliDec),
    MoodUpdate_Dec is (1-MoodUpdate_AliDec).
```

Neste exercício é-nos pedido o cálculo das emoções Esperança/Medo e Alívio/Deceção. Como exemplo, o utilizador uA tem esperança que uB, uC e uD sejam incluídos numa sugestão de grupo de utilizadores e tem medo que uE e uF sejam incluídos nesse mesmo grupo. Na situação em estudo apenas o userB entrou enquanto dos dois indesejados entrou um.

#### SITUAÇÃO EM ESTUDO

	DESEJADOS	NÃO DESEJADOS
Total de users	3	2
NÃO ENTRARAM	2	1
ENTRARAM	1	1

No que diz respeito às emoções, a esperança e deceção estão ligados aos utilizadores desejados enquanto o alívio e medo refere-se aos utilizadores indesejados.

	DESEJADOS	NÃO DESEJADOS
NÃO ENTRARAM	Dececao	Alivio
ENTRARAM	Esperanca	Medo

Numa primeira parte fizemos os cálculos referentes à esperança e à decepção. Se observarmos que existem mais utilizadores desejados que entraram do que aqueles que ficaram excluídos temos um aumento da esperança e uma diminuição da decepção. Numa segunda parte fazemos os cálculos referentes ao alívio e medo. Se verificarmos um maior número de utilizadores indesejados no grupo sugerido do que os excluídos teremos, como resultado, um aumento do medo e uma redução de alívio.

No exemplo acima temos, numa primeira fase, uma diminuição da esperança e um aumento de decepção. Em relação aos utilizadores não desejados, temos um empate no número de utilizadores que entraram e os que ficaram de fora pelo que esta situação não altera o valor da emoção.

```
| moodUpdate_EM_AD('Sarah Silva', ['Pedro Mourao', 'Rafael Soares', 'Rui Mariske'], ['Rafael Soares', 'Carlos Moutinho', 'Dario Ornelas'],  
['Rui Mariske', 'Jonatas Granjeiro'], Esperanca, Medo, Alivio, Dececao).  
Esperanca = 0.3333333333333337,  
Medo = 0.6666666666666666,  
Alivio = 0.1666666666666669,  
Dececao = 0.8333333333333333 .
```



## PARA AS EMOÇÕES ORGULHO, REMORSO, GRATIDAO,RAIVA CONSIDERE O FACTO DE UM DADO UTILIZADOR TER OU NÃO TER UM DADO TAG

```

moodUpdate_OR_GR(User,NumCommonTags,MinNumberUsers,MandatoryListTags,ListWantedUsers,ListUnwantedUsers,MoodUpdate_Org,MoodUpdate_Rem,
MoodUpdate_Gra,MoodUpdate_Rai):-

    suggestGroups(User,NumCommonTags,MinNumberUsers,MandatoryListTags, ListSuggestion),
    suggestGroups(User,NumCommonTags,MinNumberUsers,[],ListSuggestion_NoTags),!,
    get_UserMood_OrgRem(User, UserMood_OrgRem),
    get_UserMood_GraRai(User, UserMood_GraRai),
    get_UsersIncluded(ListSuggestion_NoTags,ListSuggestion, ListNewUsersIncluded),
    get_UsersExcluded(ListSuggestion_NoTags,ListSuggestion, ListNewUsersExcluded),
    get_WantedUsers_Included(ListNewUsersIncluded,ListWantedUsers,ListWanted_Included),
    get_WantedUsers_Excluded(ListNewUsersExcluded,ListWantedUsers,ListWanted_Excluded),
    get_UnwantedUsers_Included(ListNewUsersIncluded,ListUnwantedUsers,ListUnwanted_Included),
    get_UnwantedUsers_Excluded(ListNewUsersExcluded,ListUnwantedUsers,ListUnwanted_Excluded),

    %Variaveis
    length(ListWantedUsers,A), Num_AllWantedUsers is A,
    length(ListUnwantedUsers,B), Num_AllUnwantedUsers is B,
    length(ListWanted_Included,C),Num_WantedUsers_Included is C,
    length(ListWanted_Excluded,D),Num_WantedUsers_Excluded is D,
    length(ListUnwanted_Included,E),Num_UnwantedUsers_Included is E,
    length(ListUnwanted_Excluded,F),Num_UnwantedUsers_Excluded is F,

    %Media
    Media_OrgRai is (Num_AllWantedUsers/2),
    Media_GraRem is (Num_AllUnwantedUsers/2),

    %Ratios (V/Max.V)
    Orgulho_Ratio is (Num_WantedUsers_Included/Num_AllWantedUsers),
    Remorso_Ratio is (Num_UnwantedUsers_Included/Num_AllUnwantedUsers),
    Gratidao_Ratio is (Num_UnwantedUsers_Excluded/Num_AllUnwantedUsers),
    Raiva_Ratio is (Num_WantedUsers_Excluded/Num_AllWantedUsers),

    % Calculos Part I: Orgulho e Raiva
    % Orgulho
    ( (Num_WantedUsers_Included > Media_OrgRai, moodUpdate_Up(UserMood_OrgRem,Orgulho_Ratio, MoodUpdate_OrgRem_1));
      (Num_WantedUsers_Included < Media_OrgRai, moodUpdate_Down(UserMood_OrgRem,Orgulho_Ratio, MoodUpdate_OrgRem_1));
      (MoodUpdate_OrgRem_1 is UserMood_OrgRem)),
    % Raiva
    ( (Num_WantedUsers_Excluded > Media_OrgRai, moodUpdate_Down(UserMood_GraRai,Raiva_Ratio, MoodUpdate_GraRai_1));
      (Num_WantedUsers_Excluded < Media_OrgRai, moodUpdate_Up(UserMood_GraRai,Raiva_Ratio,MoodUpdate_GraRai_1));
      (MoodUpdate_GraRai_1 is UserMood_GraRai)),

    % Calculos Part II: Gratidao e Remorso
    % Gratidao
    ( (Num_UnwantedUsers_Excluded > Media_GraRem, moodUpdate_Up(MoodUpdate_GraRai_1,Gratidao_Ratio,MoodUpdate_GraRai));
      (Num_UnwantedUsers_Excluded < Media_GraRem, moodUpdate_Down(MoodUpdate_GraRai_1,Gratidao_Ratio,MoodUpdate_GraRai));
      (MoodUpdate_GraRai is MoodUpdate_GraRai_1)),

    % Remorso
    ( (Num_UnwantedUsers_Included > Media_GraRem, moodUpdate_Down(MoodUpdate_OrgRem_1,Remorso_Ratio,MoodUpdate_OrgRem));
      (Num_UnwantedUsers_Included < Media_GraRem, moodUpdate_Up(MoodUpdate_OrgRem_1,Remorso_Ratio,MoodUpdate_OrgRem));
      (MoodUpdate_OrgRem is MoodUpdate_OrgRem_1)),

    % ver os valores por emocao
    MoodUpdate_Org is (MoodUpdate_OrgRem),
    MoodUpdate_Rem is (1-MoodUpdate_OrgRem),
    MoodUpdate_Gra is (MoodUpdate_GraRai),
    MoodUpdate_Rai is (1-MoodUpdate_GraRai).

```

Neste exercício é-nos pedido o cálculo das emoções Orgulho/Remorso e Gratidão/Raiva tendo em conta a inclusão ou não de tags. O Orgulho e o Remorso estão relacionados com o facto de termos incluído tags que fizeram com que determinado utilizador pretendido/indesejado tenha sido incluído na sugestão de grupo. Já a Gratidão e Raiva estão ligados com a exclusão de um determinado utilizador do grupo.

	DESEJADOS	NÃO DESEJADOS
NÃO ENTRARAM	Raiva	Gratidao
ENTRARAM	Orgulho	Remorso

## Exemplo

Tag obrigatório: travel

sugestão de grupos sem tag associada: [Rui, Carlos, Jonatas, Fabio, Dario]

sugestão de grupo com tag associada: [Pedro, Fabio, Dario, Rafael]

Como visto acima, a presença de uma tag obrigatória fez com que alguns utilizadores fossem inseridos e outros excluídos o que vai levar a modificações no estado emocional do utilizador.

Para responder a este requisito, começamos por obter o grupo sugerido se não houvesse tags obrigatório e outro grupo com tags obrigatórias. Comparando ambas as listas vamos criar 2 listas: lista com utilizadores adicionados devido a presença da tag e uma segunda lista com os utilizadores excluídos devido à mesma tag. De seguida vamos fazer uma interceção entre listas para obter os utilizadores desejados que foram incluídos, os utilizadores desejados que foram excluídos, os utilizadores não desejados que foram incluídos e os utilizadores não desejados que foram excluídos. Esta repartição pode ser visualizada na imagem abaixo:

### SITUAÇÃO EM ESTUDO

-	DESEJADOS	NÃO DESEJADOS
Total de users	3	2
NÃO ENTRARAM	2	1
ENTRARAM	1	1

Numa primeira parte fizemos os cálculos referentes ao orgulho e raiva. Se observarmos que existem mais utilizadores desejados que entraram do que aqueles que ficaram excluídos temos um aumento do orgulho e uma diminuição da raiva. Numa segunda parte fazemos os cálculos referentes à gratidão e remorsos. Se verificarmos um maior número de utilizadores indesejados no grupo sugerido do que os excluídos teremos, como resultado, um aumento do remorso e uma redução de orgulho.

No exemplo acima temos, numa primeira fase, uma diminuição do orgulho e um aumento da raiva. Em relação aos utilizadores não desejados, temos um empate no número de utilizadores que entraram e os que ficaram de fora pelo que esta situação não altera o valor da emoção.

```
?- moodUpdate_OR_GR('Sarah Silva', 2,2,['travel'],['Carlos Moutinho', 'Jonatas Granjeiro',  
'Pedro Mourao'],['Rui Mariske', 'Rafael Soares'],Orgulho,Remorsos,Gratidao,Raiva).  
Orgulho = 0.33333333333333337,  
Remorsos = 0.66666666666666666,  
Gratidao = 0.16666666666666669,  
Raiva = 0.8333333333333333 .
```

## ESTADO DA ARTE DO USO DE METODOLOGIAS/TECNOLOGIAS (LÍNGUA NATURAL) APLICADAS AO PROBLEMA DE TRATAMENTO DE EMOÇÕES EM REDES SOCIAIS DE NOVOS VALORES DOS ESTADOS EMOCIONAIS

A língua natural, é a área da Inteligência Artificial que procura compreender textualmente, linguagem escrita por os humanos. Sendo a entrada do sistema em formato textual não existe a necessidade de utilizar tecnologias como reconhecimento de fala ou reconhecimento ótico de caracteres. Este tipo de tecnologia tem uma forte utilidade nos dias de hoje pois conseguimos ler, tratar e arquivar informação e com ela extrapolar dados. Tarefa simples para os seres humanos, mas bastante complicada para os sistemas computacionais. Para isso, o software tem de interpretar a sintaxe e a semântica das frases analisadas (Babu, 2022).

O uso de ontologia de domínio para extrair o tema e o atributo de frases e obter a polaridade emocional do tema ao combinar análise sintática é uma das formas eficazes de distinguir a subjetividade da frase e é feita através do uso de adjetivos. Quando as tendências emocionais são extraídas e identificadas num método não supervisionado, as unidades de expressão emocional ocultas referem-se aos advérbios, substantivos e adjetivos no texto (Zhoua, Xuc, & Yend, 2019).

Atualmente, as emoções que são atribuídas e que circulam nas redes sociais podem ser uma ferramenta extremamente poderosa, pois com esses dados consegue-se retirar algumas conclusões. Onde se podem observar esses casos é nas redes sociais associadas a plataformas de *e-commerce*. A satisfação e a confiança são determinadas pela experiência cognitiva e experiência emocional de compras online. Estes afetam a intenção de recompra dos consumidores que intensificam mais o comportamento de recompra do consumidor, levando assim a fortes intenções de recompra no futuro (Zhoua, Xuc, & Yend, 2019).

As avaliações online refletem o estado psicológico das pessoas e emoção abrangente dos consumidores on-line em relação aos produtos. Com isto, tornar os consumidores compradores recorrentes é um dos maiores desafios para os fornecedores de rede (Zhoua, Xuc, & Yend, 2019).

Com o aumento do número de avaliações online, a extração deste tipo de avaliação está a surgir gradualmente. A tendência subjetiva dos autores pode ser observada nos comentários e a extração dos mesmos que reflete a atitude do autor, é chamado de extração de opinião e envolve principalmente os seguintes aspetos: localização subjetiva da frase, extração de recursos, extração de emoção do utilizador, polaridade emocional, julgamento e visualização dos resultados da extração (Zhoua, Xuc, & Yend, 2019).

Portanto, as duas dimensões de satisfação e confiança dos consumidores são calculadas através da extração de avaliações online (Zhoua, Xuc, & Yend, 2019).

Embora estejamos a falar sobre um caso bastante específico, podemos aplicar esta metodologia de tratamento de emoções para detetar sentimentos e emoções de tweets e as suas respostas. Com isso, medir as pontuações de influência de utilizadores com base em vários parâmetros de utilizadores e *tweets*. Por fim, podemos usar estas informações para gerar recomendações generalizadas e personalizadas para utilizadores, com base na sua atividade no *Twitter* (Alshammari & AlMansour, 2019).

A *Sentiment Analysis* também conhecida por “*Opinion Mining*”, trata-se do estudo computacional de opiniões, atitudes e emoções relativo a uma entidade (Medhat, Hassan, & Korashy, 2014).

Denota-se também uma ligeira diferença entre “*Sentiment Analysis*” e “*Opinion Mining*”. Segundo Krishnan Mubashir et al., “*Sentiment Analysis*” é definido como uma atitude, pensamento ou julgamento motivado por alguma sensação, enquanto “*Opinion Mining*” é definido como um ponto de vista, julgamento ou visão formada sobre um assunto em particular. Ou seja, enquanto uma opinião denota uma visão concreta sobre algo, um sentimento é parte de uma sensação expressa (KrishnanMubashir, BabaGurinder, & Mariappan, 2021).

Tomando por base a definição acima apresentada, o “*Sentiment Analysis*” procura identificar sentimentos expressos, por exemplo em opiniões, e posteriormente classificá-los quanto à sua polarização (KrishnanMubashir, BabaGurinder, & Mariappan, 2021).

Segundo Liu, os sentimentos podem ser classificados da seguinte forma:

- “*Word Level*”: Na classificação do sentimento, a palavra é categorizada através da utilização da polarização. A classificação neste nível é atribuída através de metodologias baseadas em dicionários, atribuindo assim uma classificação a palavras, como por exemplo “Feliz” ou “contente”, uma alta conotação de felicidade.
- “*Sentence Level*”: Refere-se à classificação da frase, quanto à sua subjetividade ou objetividade e posterior conotação, consoante a sua classificação. Neste nível de classificação é possível usar não apenas texto, mas outros meios de contextualização como “Emoticons”, ajudando assim na exatidão da atribuição de uma classificação positiva ou negativa.
- “*Document Level*”: Este nível é efetuado em documentos, como artigos ou notícias, dando uma avaliação ao documento um sentimento negativo, positivo ou neutro,

tendo por base a conotação geral expressa no mesmo sob análise. Neste caso, outros meios de expressão como *Emoticons* também ajudam na atribuição da classificação geral.

– “*Feature/Aspect Level*”: Nesta análise, primeiramente são caracterizados os objetos em análise e posteriormente é efetuada a análise da opinião expressa, realizando uma análise mais granular, aumentando assim a exatidão da atribuição da classificação. (Liu, 2020)

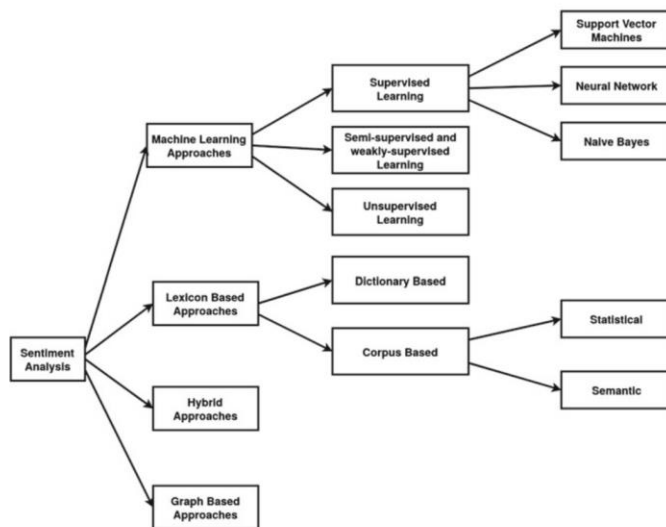


Figura 1- Análise em redes sociais: técnicas, conjuntos de dados e questões em aberto.

Fonte: (KrishnanMubashir, BabaGurinder, & Mariappan, 2021)

Para análise do sentimento transmitido, referem-se as seguintes técnicas:

– “*Knowledge/Lexicon Based*”: Inclui a classificação de texto através de palavras que provêm de emoções (Moreo, Romero, Castro, & Zurita, 2012).

– “*Statistical/ML*”: Usando algoritmos de *Machine Learning* como “*latent semantic analysis*” e “*deep learning*” para melhorar a precisão na detecção de sentimentos (Müller & Metternicha, 2021).

– “*Hybrid*”: Integrando as duas técnicas anteriores, torna possível uma melhoria significativa na interpretação e classificação da análise sentimental (Sahinidis, 2018).

– “*Graph Based*”: São estruturas baseadas em grafos sobre opiniões para extrair assim determinadas características, correspondentes aos nós do grafo apresentado (Bourbakis, 2014).

## BIBLIOGRAFIA

- Alshammari, N. F., & AlMansour, A. A. (2019). State-of-the-art review on Twitter Sentiment Analysis. *IEEE Xplore*.
- Babu, N. K. (2022). Sentiment Analysis in Social Media Data for Depression Detection Using Artificial Intelligence: A Review. *SN COMPUT. SCI*.
- Bourbakis, M. T. (2014). Graph-Based Methods for Natural Language Processing and Understanding—A Survey and Analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 59-71.
- KrishnanMubashir, C., BabaGurinder, M., & Mariappan, S. (2021). Viral Marketing: A New Horizon and Emerging Challenges. *Springer Link*.
- Liu, B. (2020). *Sentiment Analysis*. University of Illinois, Chicago: Cambridge University Press.
- Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*.
- Moreo, A., Romero, M., Castro, J., & Zurita, J. (2012). Lexicon-based Comments-oriented News Sentiment Analyzer system. *Expert Systems with Applications*, 9166-9180.
- Müller, M., & Metternich, J. (2021). Production specific language characteristics to improve NLP applications on the shop floor. *Procedia CIRP*, 1890-1895.
- Sahinidis, A. K. (2018). A hybrid LP/NLP paradigm for global optimization relaxations. *Mathematical Programming Computation*, 383–421.
- Zhoua, Q., Xuc, Z., & Yend, N. Y. (2019). User sentiment analysis based on social network information and its application in consumer reconstruction intention. *ELSEVIER*.