

Ponto de situação

25/03/20

Recuperação e Identificação de momentos em Imagens/Vídeos.

Objetivos Propostos a 18/03/20

1. Testar tópicos do imageclef 2019 e aperfeiçoar o algoritmo. **(CONCLUÍDO)**
 - a. Ainda falta testar um 2º PDF de 2019.
2. Melhorar sistema de “negative words”. **(CONCLUÍDO)**
3. Recuperar imagens consoante várias condições. **(AINDA POR FAZER)**
 - a. Algumas dúvidas neste ponto (Secção de dúvidas) -> **PAGINA 24**
4. Não recuperar imagens que tenham nas features “Negative Words”
(CONCLUÍDO)

Novas implementações no trabalho

1. O ficheiro que trata de processar o texto gera agora um ficheiro json, onde são guardadas todas as features relativamente as features obtidas do texto.
2. O processamento de texto passou a ser realizado com o modelo médio do spacy.
3. O small_data.json foi aumentado de 20.000 para 100.000 linhas, que representa cerca de 1000 imagens. Isto leva a 5 vezes mais o tempo de computação que anteriormente, mas permite testar mais a fundo o algoritmo.
 - a. Small_data.json é um ficheiro mais pequeno do ficheiro original de 10.000.000 linhas que representa a totalidade das 200.000 imagens.

Exemplo do json gerado pelo
algoritmo de processamento de
texto criado:

```
{
  "relevant things": [
    "public transports",
    "transports",
    "home",
    "form",
    "public transport",
    "transport",
    "bus",
    "taxi",
    "train",
    "boat"
  ],
  "activities": [
    "using public transports"
  ],
  "dates": [
    "2015",
    "2018"
  ],
  "locations": [
    "home country",
    "ireland"
  ],
  "other words": [
    "public",
    "ul",
    "taking",
    "considered",
    "relevant",
    "driving"
  ],
  "other things": [
    "country",
    "moments",
    "car"
  ],
  "inside": "Not known",
  "outside": "Not known",
  "location or thing": [],
  "negative relevant thing": [
    "car"
  ],
  "negative activities": [
    "driving a car"
  ],
  "negative locations": [],
  "negative dates": []
}
```

Teste de tópicos do imageclef 2019
e
aperfeiçoação do algoritmo

1. O algoritmo criado para os tópicos de 2020 foi testado agora para os tópicos de 2019, na esperança de obter resultados decentes para tópicos nunca antes testados.
 - a. **O objetivo é ter o algoritmo o melhor preparado possível para o segundo pdf de tópicos de 2020 que irá sair do imageclef.**
2. Apesar de ter sido necessário alguns ajustes para apenas o tópico 8 de 2019, penso que os resultados foram bons.
3. Ainda falta testar para um segundo PDF de 2019.

<title>**Icecream by the Sea**</title>

<description>Find the moment when ul was eating an icecream beside the sea.</description>

<narrative>To be relevant, the moment must show both the ice cream with cone in the hand of ul as well as the sea clearly visible. Any moments by the sea, or eating an ice cream which do not occur together are not considered to be relevant.</narrative>

</topic>

<topic>

```
{
  "relevant things": [
    "icecream",
    "ice cream",
    "cream",
    "cone"
  ],
  "activities": [
    "eating an icecream"
  ],
  "dates": [],
  "locations": [
    "sea",
    "hand"
  ],
  "other words": [
    "ul",
    "eating",
    "relevant",
    "clearly",
    "visible",
    "occur",
    "considered"
  ],
  "other things": [
    "moment",
    "ice",
    "moments"
  ],
  "inside": false,
  "outside": true,
  "location or thing": [],
  "negative relevant thing": [],
  "negative activities": [],
  "negative locations": [],
  "negative dates": []
}
```

<title>**Having food in a restaurant**</t

itle>

<description>Find the moment when ul was eating food or drinking in a restaurant.</description>

<narrative>Ul was eating food in a restaurant while away from home. Any kinds of dishes are relevant. Only Drinking coffee and have dessert in a cafe won't be relevant</narrative>

</topic>

<topic>

```
{
  "relevant things": [
    "food",
    "coffee",
    "dessert",
    "home",
    "dishes"
  ],
  "activities": [
    "having food",
    "eating food",
    "drinking coffee"
  ],
  "dates": [],
  "locations": [
    "restaurant",
    "cafe"
  ],
  "other words": [
    "having",
    "ul",
    "eating",
    "away",
    "relevant",
    "wo"
  ],
  "other things": [
    "drinking",
    "kinds"
  ],
  "inside": true,
  "outside": false,
  "location or thing": [],
  "negative relevant thing": [],
  "negative activities": [],
  "negative locations": [],
  "negative dates": []
}
```


<title>**Watching Videos**</title>

<description>Find the moment when u1 was watching video when using other digital devices.</description>

<narrative>To be relevant, u1 must be watching videos in any location and any digital devices can be considered. For example: TV machine, tablet, mobile phone, laptop, desktop computer. </narrative>

</topic>

```
"relevant things": [
  "videos",
  "video",
  "digital devices",
  "devices",
  "tv machine",
  "mobile phone",
  "phone",
  "desktop computer",
  "example",
  "tablet",
  "laptop"
],
"activities": [
  "watching videos",
  "watching video"
],
"dates": [],
"locations": [
  "location"
],
"other words": [
  "watching",
  "u1",
  "digital",
  "relevant",
  "considered",
  "mobile"
],
"other things": [
  "tv",
  "machine",
  "desktop",
  "computer"
],
"inside": true,
"outside": false,
"location or thing": [],
"negative relevant thing": [],
"negative activities": [],
"negative locations": [],
"negative dates": []
```

<uid>u1</uid>

<title>**Photograph of a Bridge**</title>

<description>Find the moment when u1 was taking a photo of a bridge.</description>

<narrative>U1 was walking on a pedestrian street and stopped to take a photo of a bridge. Moments when u1 was walking on a street without stopping to take a photo of a bridge are not relevant. Any other moment showing a bridge when a photo was not being taken are also not considered to be relevant.</narrative>

```
{
  "relevant things": [
    "photo",
    "bridge",
    "pedestrian street",
    "bridge"
  ],
  "activities": [
    "taking a photo",
    "walking on a pedestrian"
  ],
  "dates": [],
  "locations": [],
  "other words": [
    "u1",
    "taking",
    "walking",
    "stopped",
    "stopping",
    "relevant"
  ],
  "other things": [
    "photograph",
    "bridges",
    "pedestrian",
    "moments"
  ],
  "inside": "Unknown",
  "outside": "Unknown",
  "location or thing": [
    "pedestrian street",
    "street"
  ],
  "negative relevant thing": [],
  "negative activities": [],
  "negative locations": [],
  "negative dates": []
}
```

```
<id>005</id>
<type>precision</type>
<uid>u1</uid>
<title>Grocery shopping</title>
<description>Find the moment when u1 was shopping for
food in a grocery shop.</description>
<narrative>To be considered relevant, u1 must be clearly
in a grocery shop and bought something from the it</narrative>
</topic>
<topic>
```

```
1  {
2    "relevant things": [
3      "food"
4    ],
5    "activities": [
6      "shopping for food"
7    ],
8    "dates": [],
9    "locations": [
10     "grocery shop"
11   ],
12   "other words": [
13     "shopping",
14     "u1",
15     "clearly",
16     "bought"
17   ],
18   "other things": [
19     "grocery",
20     "shop"
21   ],
22   "inside": true,
23   "outside": false,
24   "location or thing": [],
25   "negative relevant thing": [],
26   "negative activities": [],
27   "negative locations": [],
28   "negative dates": []
29 }
```

<title>**Playing a Guitar**</title>

<description>Find the moment when U1 or a man is playing guitar in view.</description>

<narrative> Any use of guitars indoors could be considered relevant. Any type of Guitar could be considered as relevant.</narrative>

</topic>

<topic>

```
{
  "relevant things": [
    "guitar",
    "guitars"
  ],
  "activities": [
    "playing a guitar",
    "playing guitar"
  ],
  "dates": [],
  "locations": [
    "view"
  ],
  "other words": [
    "playing",
    "u1",
    "indoors",
    "considered",
    "relevant"
  ],
  "other things": [
    "man",
    "use",
    "type"
  ],
  "inside": true,
  "outside": false,
  "location or thing": [],
  "negative relevant thing": [],
  "negative activities": [],
  "negative locations": [],
  "negative dates": []
}
```

```
<title>Cooking</title>
<description>Find moments when ul was cooking
food.</description>
<narrative>The moments shows U1 was cooking food at any
places are relevant</narrative>
```

```

{
  "relevant things": [
    "food"
  ],
  "activities": [
    "cooking food"
  ],
  "dates": [],
  "locations": [],
  "other words": [
    "u1",
    "shows",
    "relevant"
  ],
  "other things": [
    "cooking",
    "places"
  ],
  "inside": "Unknown",
  "outside": "Unknown",
  "location or thing": [],
  "negative relevant thing": [],
  "negative activities": [],
  "negative locations": [],
  "negative dates": []
}
```

Topico 8 - precisou de ajustes

<title>**Car Sales Showroom**</title>

<description>Find the moments when ul was in a car sales showroom.</description>

<narrative>ul visited a car sales showroom a few times. Relevant moments show ul indoors in a car sales showroom, either looking at cars or waiting for a salesman sitting at a table. Any moments looking at cars while outside of a showroom are not considered relevant.</narrative>

```
{
  "relevant things": [
    "car sales",
    "cars",
    "salesman"
  ],
  "activities": [
    "visited a car sales showroom",
    "looking at cars",
    "waiting for a salesman"
  ],
  "dates": [],
  "locations": [
    "sales showroom",
    "car sales showroom",
    "showroom",
    "table"
  ],
  "other words": [
    "ul",
    "visited",
    "looking",
    "waiting",
    "sitting",
    "outside",
    "considered",
    "relevant"
  ],
  "other things": [
    "car",
    "sales",
    "times",
    "indoors",
    "moments"
  ],
  "inside": true,
  "outside": false,
  "location or thing": [],
  "negative relevant thing": [],
  "negative activities": [],
  "negative locations": [],
  "negative dates": []
}
```

<title>**Paper or book reviewing**</title>

<description>Find all moments when u1 was reading a paper or book.</description>

<narrative>To be relevant, the paper or book must be visible in front of U1 and sometimes U1 use a pen to mark on the paper or book </narrative>

</topic>

```
{
  "relevant things": [
    "paper",
    "pen",
    "book"
  ],
  "activities": [
    "reading a paper",
    "mark on the paper"
  ],
  "dates": [],
  "locations": [],
  "other words": [
    "reviewing",
    "u1",
    "reading",
    "relevant",
    "visible",
    "use",
    "mark"
  ],
  "other things": [],
  "inside": "Unknown",
  "outside": "Unknown",
  "location or thing": [
    "paper"
  ],
  "negative relevant thing": [],
  "negative activities": [],
  "negative locations": [],
  "negative dates": []
}
```

Melhoramento de sistema “Negative Words”

Melhoramento idealizado :

```
669 ##### USO DAS PALAVRAS NEGATIVAS #####
670 |
671 |
672 for thing in positive_relevant_things:
673     for negative_thing in negative_relevant_things:
674         if nlp(thing).similarity(nlp(negative_thing)) > 0.8:
675             del negative_relevant_things[negative_relevant_things.index(negative_thing)]
676 |
677 for activity in positive_activities:
678     for negative_activity in negative_activities:
679         if nlp(activity).similarity(nlp(negative_activity)) > 0.8:
680             del negative_activities[negative_activities.index(negative_activity)]
681 |
682 for loc in positive_locations:
683     for negative_location in negative_locations:
684         if nlp(loc).similarity(nlp(negative_location)) > 0.8:
685             del negative_locations[negative_locations.index(negative_location)]
686 |
687 for date in positive_dates:
688     for negative_date in negative_dates:
689         if nlp(date).similarity(nlp(negative_date)) > 0.8:
690             del negative_dates[negative_dates.index(negative_date)]
691 |
692 |
693 |
```

Se coisas negativas forem muito idênticas a coisas positivas, são positivas, caso contrário são negativas.

Problema demonstrado no relatório passado

```
..... POSITIVE .....
Relevant things ['car repair', 'car', 'gloves', 'hand', 'hammer', 'phone']
Activities ['repairing his car']
Dates []
Locations: ['garden']
Location or Thing ['hand']
User inside: True
User outside: False
..... NEGATIVE .....
Negative Relevant thing ['cars', 'repair action', 'action']
Activities ['showing repair action']
Dates []
Locations: []
..... RANDOM .....
Other words: ['ui', 'repairing', 'held', 'considered', 'relevant', 'showing']
other things ['repair', 'moments', 'cars', 'action']
.....
```

Nesta situação, tanto a palavra “cars”, como a atividade “showing repair action” são positivas, no entanto, como não foram detetadas anteriormente no texto, ele considera-as negativas.

Uma **solução** idealizada, é correr a função `similarity()` do SpaCy entre todas as palavras positivas e negativas, desta forma “cars” vai acusar uma similaridade muito alta com “car” e vai passar a palavra positiva.

Para a situação da atividade, a ainda não existe nenhuma solução idealizada.

Atual situação com o melhoramento implementado :

```
{
  "relevant things": [
    "car repair",
    "car",
    "gloves",
    "hand",
    "hammer",
    "phone"
  ],
  "activities": [
    "repairing his car"
  ],
  "dates": [],
  "locations": [
    "garden"
  ],
  "other words": [
    "ul",
    "repairing",
    "held",
    "considered",
    "relevant",
    "showing"
  ],
  "other things": [
    "repair",
    "moments",
    "cars",
    "action"
  ],
  "inside": true,
  "outside": false,
  "location or thing": [
    "hand"
  ],
  "negative relevant thing": [
    "repair action",
    "action"
  ],
  "negative activities": [
    "showing repair action"
  ],
  "negative locations": [],
  "negative dates": []
}
```

“Cars” já não é considerado como “negative relevant thing”.

No entanto, “showing repair action” ainda está errado em ser considerada como “negative relevant thing” e ainda requer a idealização de uma solução.

Não recuperar imagens que tenham nas features
“Negative Words”

Algoritmo:

```
71 ##### MAIN #####
72
73 if __name__ == '__main__':
74
75     dir_path = os.path.dirname(os.path.realpath(__file__))
76     text_data = dir_path + "/NLP_data.json"
77     images_data = dir_path + "/small_data.json"
78
79     text_data = json.loads(open(text_data).read())
80     images_data = json.loads(open(images_data).read())
81
82     nlp = spacy.load("en_core_web_md") # make sure to use larger model
83
84     returned_images = []
85
86     # Find positives
87     for image_name in images_data.keys():
88         print(image_name)
89         returned_images = find_image(images_data, text_data, returned_images, image_name, "concepts", "relevant things")
90         returned_images = find_image(images_data, text_data, returned_images, image_name, "location", "locations")
91         returned_images = find_image(images_data, text_data, returned_images, image_name, "activity", "activities")
92         returned_images = find_image(images_data, text_data, returned_images, image_name, "local_time", "dates")
93
94     ## Find Negatives
95     for image_name in returned_images:
96         print(image_name)
97         returned_images = find_negative(images_data, text_data, returned_images, image_name, "concepts", "negative relevant thing")
98         returned_images = find_negative(images_data, text_data, returned_images, image_name, "location", "negative locations")
99         returned_images = find_negative(images_data, text_data, returned_images, image_name, "activity", "negative activities")
100         returned_images = find_negative(images_data, text_data, returned_images, image_name, "local_time", "negative dates")
101
102
103     print(returned_images)
104
105
```

No primeiro “for” , o algoritmo procura imagens com features idêntica às que foram obtidas pelo processamento de texto, guardando sempre no array “returned_images”.

No segundo “for” as imagens que foram guardadas no array anterior são novamente corridas de forma a procurar por “negative words”, caso haja uma detecção, essa imagem é apagada do array “returned_images”

Isto permite poupar tempo, pois se fosse corrido as “negative words” para todas as imagens do small_data.json, o algoritmo iria demorar muito mais tempo a correr.

```
def find_negative (images_data, text_data, returned_images,image_name, json_image_key,json_text_key):

    if json_image_key == "concepts":
        for concept in images_data[image_name][json_image_key].keys():
            image_info = concept
            if image_name not in returned_images: break

        for negative_data in text_data[json_text_key]:

            if image_name not in returned_images : break

            if nlp(negative_data).similarity(nlp(image_info)) >= 0.8:
                del returned_images[returned_images.index(image_name)]
                print("Negative picture", image_name)
                return returned_images
        return returned_images

    if json_image_key == "local time":
        image_info = images_data[image_name][json_image_key][0:4]
    else:
        image_info = images_data[image_name][json_image_key]

    for negative_data in text_data[json_text_key]:

        if image_name not in returned_images : break

        if nlp(negative_data).similarity(nlp(image_info)) >= 0.8:
            del returned_images[returned_images.index(image_name)]
            print("Negative picture", image_name)
    return returned_images
```

```
def find_image (images_data, text_data, returned_images,image_name, json_image_key,json_text_key):
    sim_score = 0.8

    if json_image_key == "concepts":
        for concept in images_data[image_name][json_image_key].keys():
            image_info = concept
            for data in text_data[json_text_key]:

                if nlp(data).similarity(nlp(image_info)) >= sim_score and image_name not in returned_images:
                    returned_images.append(image_name)
                    print("Saved" , image_name)
                    return returned_images

        return returned_images

    if json_image_key == "local time":
        image_info = images_data[image_name][json_image_key][0:4]
        sim_score = 1

    else:
        image_info = images_data[image_name][json_image_key]

    for data in text_data[json_text_key]:
        if nlp(data).similarity(nlp(image_info)) >= sim_score and image_name not in returned_images:
            returned_images.append(image_name)
            print("Saved" , image_name)
            return returned_images

    return returned_images
```


Resultado :

```
['b00000001 2116bq 20150223 070808e.jpg', 'b00000002 2116bq 20150223 070809e.jpg', 'b00000003 2116bq 20150223 070810e.jpg', 'b00000004 2116bq 20150223 070810e.jpg', 'b00000005 2116bq 20150223 070811e.jpg', 'b00000006 2116bq 20150223 070812e.jpg', 'b00000007 2116bq 20150223 070813e.jpg', 'b00000008 2116bq 20150223 070813e.jpg', 'b00000009 2116bq 20150223 070814e.jpg', 'b00000010 2116bq 20150223 070822e.jpg', 'b00000011 2116bq 20150223 070859e.jpg', 'b00000012 2116bq 20150223 070931e.jpg', 'b00000013 2116bq 20150223 071008e.jpg', 'b00000014 2116bq 20150223 071046e.jpg', 'b00000015 2116bq 20150223 071119e.jpg', 'b00000016 2116bq 20150223 071152e.jpg', 'b00000017 2116bq 20150223 071228e.jpg', 'b00000018 2116bq 20150223 071305e.jpg', 'b00000019 2116bq 20150223 071344e.jpg', 'b00000020 2116bq 20150223 071416e.jpg', 'b00000021 2116bq 20150223 071451e.jpg', 'b00000022 2116bq 20150223 071528e.jpg', 'b00000023 2116bq 20150223 071607e.jpg', 'b00000024 2116bq 20150223 071646e.jpg', 'b00000025 2116bq 20150223 071723e.jpg', 'b00000026 2116bq 20150223 071754e.jpg', 'b00000027 2116bq 20150223 071835e.jpg', 'b00000028 2116bq 20150223 071906e.jpg', 'b00000029 2116bq 20150223 071937e.jpg', 'b00000030 2116bq 20150223 072014e.jpg', 'b00000031 2116bq 20150223 072128e.jpg', 'b00000032 2116bq 20150223 072159e.jpg', 'b00000033 2116bq 20150223 072234e.jpg', 'b00000034 2116bq 20150223 072305e.jpg', 'b00000035 2116bq 20150223 072330e.jpg', 'b00000036 2116bq 20150223 072421e.jpg', 'b00000037 2116bq 20150223 072472e.jpg', 'b00000038 2116bq 20150223 072500e.jpg', 'b00000039 2116bq 20150223 072533e.jpg', 'b00000040 2116bq 20150223 072716e.jpg', 'b00000041 2116bq 20150223 072803e.jpg', 'b00000042 2116bq 20150223 072853e.jpg', 'b00000043 2116bq 20150223 072853e.jpg', 'b00000044 2116bq 20150223 112905e.jpg', 'b00000045 2116bq 20150223 142424e.jpg', 'b00000046 2116bq 20150223 142459e.jpg', 'b00000047 2116bq 20150223 142537e.jpg', 'b00000048 2116bq 20150223 155835e.jpg', 'b00000049 2116bq 20150223 155910e.jpg', 'b00000050 2116bq 20150223 161710e.jpg', 'b00000051 2116bq 20150223 171401e.jpg', 'b00000052 2116bq 20150223 171544e.jpg', 'b00000053 2116bq 20150223 171732e.jpg', 'b00000054 2116bq 20150223 171917e.jpg', 'b00000055 2116bq 20150223 172135e.jpg', 'b00000056 2116bq 20150223 172247e.jpg', 'b00000057 2116bq 20150223 172320e.jpg', 'b00000058 2116bq 20150223 172438e.jpg', 'b00000059 2116bq 20150223 172552e.jpg', 'b00000060 2116bq 20150223 172629e.jpg', 'b00000061 2116bq 20150223 172750e.jpg', 'b00000062 2116bq 20150223 172904e.jpg', 'b00000063 2116bq 20150223 172938e.jpg', 'b00000064 2116bq 20150223 173018e.jpg', 'b00000065 2116bq 20150223 173058e.jpg', 'b00000066 2116bq 20150223 173213e.jpg', 'b00000067 2116bq 20150223 173251e.jpg', 'b00000068 2116bq 20150223 173425e.jpg', 'b00000069 2116bq 20150223 173557e.jpg', 'b00000070 2116bq 20150223 173751e.jpg', 'b00000071 2116bq 20150223 173828e.jpg', 'b00000072 2116bq 20150223 173901e.jpg', 'b00000073 2116bq 20150223 174011e.jpg', 'b00000074 2116bq 20150223 174046e.jpg', 'b00000075 2116bq 20150223 174124e.jpg', 'b00000076 2116bq 20150223 174233e.jpg', 'b00000077 2116bq 20150223 174310e.jpg', 'b00000078 2116bq 20150223 174349e.jpg', 'b00000079 2116bq 20150223 174430e.jpg', 'b00000080 2116bq 20150223 174539e.jpg', 'b00000081 2116bq 20150223 174606e.jpg', 'b00000082 2116bq 20150223 174648e.jpg', 'b00000083 2116bq 20150223 174834e.jpg', 'b00000084 2116bq 20150223 174941e.jpg', 'b00000085 2116bq 20150223 175050e.jpg', 'b00000086 2116bq 20150223 175206e.jpg', 'b00000087 2116bq 20150223 175356e.jpg', 'b00000088 2116bq 20150223 175428e.jpg', 'b00000089 2116bq 20150223 175544e.jpg', 'b00000090 2116bq 20150223 175656e.jpg', 'b00000091 2116bq 20150223 175728e.jpg', 'b00000092 2116bq 20150223 175805e.jpg', 'b00000093 2116bq 20150223 175819e.jpg', 'b00000094 2116bq 20150223 175931e.jpg', 'b00000095 2116bq 20150223 180002e.jpg']
```

```
Total images analysed : 966
Total images with positive features : 110
Total images with negative features (deleted) : 0
Total images returned: 110
```

```
"relevant things": [
  "computer",
  "personal computer",
  "pc parts",
  "uncompleted pcs",
  "pcs",
  "scratch"
],
"activities": [
  "building personal computer"
],
"dates": [],
"locations": [
  "office"
],
"other words": [
  "building",
  "personal",
  "ul",
  "built",
  "clearly",
  "/",
  "uncompleted",
  "considered",
  "relevant"
],
"other things": [
  "pc",
  "parts",
  "moments"
],
"inside": true,
"outside": false,
"location or thing": [
  "table"
],
"negative relevant thing": [],
"negative activities": [],
"negative locations": [],
"negative dates": []
```

De notar que para este exemplo são apenas analisadas features de 966 imagens (das 200.000), e no texto analisado não existe nada detetado como “negativo” para apagar fotos do array

Dúvidas

De momento, o algoritmo basta detetar uma “coisa”, um “local”, uma “atividade” ou uma “data” para considerar a imagem relevante. Mas, como fazer para situações em que é requerido que exista uma “coisa” e seja num “local” ao mesmo tempo? Ou que tenha de decorrer uma “atividade” num “local” específico?

```
<narrative>To be considered relevant, ul must be clearly at  
the office with the PC parts on the table. Any moments that the ul  
is not in the office or there are no PC parts/uncompleted PCs on  
the table are not considered relevant.</narrative>
```

```
<narrative>To be considered relevant, ul must be clearly in a  
bar. Any moments that ul drinks beers at home or outside without  
the bar view are not considered relevant.</narrative>
```

```
</topic>
```

```
<narrative>Moments when ul was at home, looking at an old  
clock, with flowers visible, with a lamp and perhaps two small  
monsters watching ul are considered relevant. One of the monsters  
might be a long rabbit. The moments without one of the  
aforementioned conditions: monsters, flowers, and old clock are  
not considered relevant.</narrative>
```

```
</topic>
```

1º exemplo : “local” + “coisa” + “coisa” (office, pc parts, table)

2º exemplo : “atividade” + “local” (drinks beers, home)

3º exemplo : “local” + “atividade” + “coisa” + “coisa + “coisa” : (home, looking at, old clock, flowers, lamp)

E noutros exemplos em que o “local” pode ser em qualquer sitio não definido no texto?

Como procurar as imagens realmente relevantes quando as condições a impor são tão diferentes umas das outras? E quando as features das imagens se encontram a “NULL”?

Outra dúvida : Como tratar de imagens em que as localidades estão por exemplo como “NULL”. Lembro-me de numa reunião se ter falado na utilização de **interpolação**.

Localidade -> NULL

```
"b000000056_21i6bq_20150223_073651e.jpg": {
  "minute_id": "20150223_0736",
  "utc_time": "UTC_2015-02-23_07:36",
  "atributes": [
    "no horizon",
    "man-made",
    "enclosed area",
    "natural light",
    "metal",
    "matte",
    "cloth",
    "working",
    "glossy",
    "indoor lighting"
  ],
  "categories": {
    "airplane_cabin": 0.226,
    "car_interior": 0.14800000000000002,
    "beauty_salon": 0.111,
    "cockpit": 0.10400000000000001,
    "gymnasium/indoor": 0.043
  },
  "concepts": {},
  "local_time": "2015-02-23_07:36",
  "timezone": "Europe/Dublin",
  "latitude": 53.38447776,
  "longitude": -6.20734461,
  "activity": "transport",
  "location": "NULL",
  "elevation": "85.0",
  "speed": "14.5",
  "heart": "NULL",
  "calories": "1.2062000036239624",
  "steps": "0.0"
```

FIM