Fábio Renato
da Cunha Veiros

Lifelog - Algoritmo para Recuperação e Identificação de Momentos em Imagem Digital

Lifelog - Moments Retrieval Algorithm

Universidade de Aveiro
2021

Fábio Renato da Cunha Veiros

Lifelog - Algoritmo para Recuperação e Identificação de Momentos em Imagem Digital

Lifelog - Moments Retrieval Algorithm

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor António Neves, Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e da Investigadora Alina Trifan.

o júri / the jury

presidente / president

Professor Doutor Sérgio Guilherme Aleixo de Matos
Professor Auxiliar em Regime Laboral da Universidade de Aveiro

vogais / examiners committee

Professor Doutor Alípio Mário Guedes Jorge
Professor Associado da Faculdade de Ciências da Universidade do Porto (arguente)

Professor Doutor António José Ribeiro Neves
Professor Auxiliar da Universidade de Aveiro (orientador)

agradecimentos / acknowledgements

Agradeço,
Aos meus avós e aos meus pais, todo o apoio e incentivo nesta caminhada.
À Diana Martins, pelo apoio, confiança e paciência.
Aos verdadeiros amigos, por nunca me deixarem desistir.

Agradeço especialmente ao meu orientador e co-orientadora, António Neves e Alina Trifan, pela disponibilidade e orientação fundamentais na realização deste trabalho.
Agradeço ao Ricardo Ribeiro pela colaboração e disponibilidade ao longo deste trabalho.

Agradeço à Universidade de Aveiro e ao Instituto de Engenharia Electrónica e Telemática de Aveiro em colaboração com o *Texas Advanced Computer Center* da *University of Texas at Austin* por me proporcionarem a possibilidade de trabalhar com tecnologia HPC preponderante para a realização do trabalho.

Palavras Chave

Lifelogging , Recuperação de Momentos, Processamento Natural de Texto, Classificação de resultados, Sistemas de Recuperação de Informação

Resumo

O aumento da variedade e quantidade de dispositivos sensoriais portáteis ocasionou um paralelo crescimento da diversidade e quantidade de dados produzidos. Hoje em dia, qualquer individuo com recurso ao *smartphone* pessoal produz uma panóplia de registos diários de momentos. Esta tipologia de dados resulta de cenários quotidianos que são registados em imagem e frequentemente detalhados com dados biométricos bem como registos de actividades, localização e tempo. Ao armazenarmos esta diversidade de dados impõe-se a questão: como identificar e recuperar um momento exacto em largos arquivos de dados? A recuperação de um momento pode atender à simples acção de revisitar um episódio longínquo, mas também pode auxiliar pessoas com problemas de memória. A aplicação de sistemas computacionais para este fim é a principal resposta. Para além de identificarem e recuperarem um momento, são aplicados com o principal objectivo de melhorar a qualidade de vida humana.

Estes factos exigem a estes sistemas uma redução de distâncias comunicacionais entre a linguagem natural e a linguagem computacional. Para tal, são constituídos por algoritmos de processamento e análise de texto que visam estabelecer uma ligação interactiva entre utilizadores e sistema.

Neste sentido, a solução proposta nesta dissertação é baseada num algoritmo que recebe e entende o momento que o utilizador descreve e tenta devolver esse instante sob a forma de imagens retiradas da base de dados do utilizador onde esse momento possa estar representado. O seu desenvolvimento passa pela aplicação de metodologias descritas no estado de arte e novas abordagens no sistema de classificação de resultados. O algoritmo é incorporado por ferramentas *NLP* que são fundamentais na comunicação entre ambas as partes. Além disso, engloba a função matemática *TFIDF* com acções de vectorização auxiliada pela similaridade de cosseno que é responsável por seleccionar os momentos que mais se identificam com a descrição do utilizador. Também a função BM25 foi introduzida no algoritmo visando reforçar a análise de similaridades entre pergunta e respostas. A coligação de ambas as técnicas atribuem ao algoritmo uma maior probabilidade na devolução do momento correcto.

O mecanismo desenvolvido mostra resultados bastante satisfatórios e interessantes uma vez que em várias interacções devolve o momento correcto ou pelo menos identifica episódios similares á descrição do utilizador.

O conhecimento adquirido ao longo desta dissertação permite-me concluir que o algoritmo teria uma maior valorização com um redobrado ênfase na descrição textual de um momento introduzida pelo utilizador. A identificação automática de campos chave, permitiria que o sistema de filtragem, aplicado no algoritmo, se tornasse totalmente automatizado.

Keywords

Lifelogging, Moments Retrieval, Natural Language Processing, Ranking, Retrieval Systems

Abstract

The increase of the variety and quantity of the wearable devices brought a parallel growth of the diversity and amount of data produced. Nowadays any individual using a personal smartphone produces a large amount of daily moments records. These data typology results from daily scenarios recorded in image and detailed with biometric data as well activities, location and time records. When storing this diversity and amount of data, a question arises: how can we identify and retrieve an exact moment in large data archives? A moment retrieval can serve the simple action of revisiting a distant episode, but it can also support a person with memory disorders. The application of computer systems for this purpose is the main answer. In addition to identifying and retrieving a moment, they are applied with the main objective of improving the quality of human life.

These facts require these systems to reduce communicational distances between natural language and computer language. Therefore, they consist of processing and text analysis algorithms that aim to establish an interactive link between the users and the system.

In this sense, the proposed solution in this dissertation is based on an algorithm that receives and understands the moment described by the user and tries to return that moment in the form of images taken from the user's database where that moment can be represented. Its development involves the application of methodologies described in the state of the art and new approaches in the results ranking system. The algorithm is incorporated by NLP tools that are fundamental in the communication between both parties. Moreover it incompasses TFIDF math function with vectorization tasks supported by cosine similarity responsible for selecting identical moments to the user description. Also the BM25 function was introduced in the algorithm aiming to reinforce the analysis of similarities between question and answers. The combination of both techniques gives the algorithm a greater probability of returning the correct moment.

The developed mechanism shows very satisfactory and interesting results, considering the fact that in several interactions they return the correct moment or at least identify similar episodes comparing to the user's description.

The knowledge acquired throughout this dissertation allows me to conclude that the algorithm would have a greater value with an emphasis on the textual moment description introduced by the user. The automatic identification of key fields would allow the filtering system, applied in the algorithm, to become fully automated.

Conteúdo

Lista de Figuras

Lista de Tabelas

Glossário

NLP	Natural Language Processing
API	Aplication Programming Interface
CSV	Comma separated values
JSON	JavaScript Object Notation
LSC	Lifelog Search Challenge
RS	Retrieval System
VBS	Video Browser Showdown
CNN	Convolutional Neural Network
NTCIR	(National Institute of Informatics) Test Collection for Information Resources
TFIDF	Term Frequency Inverse Document Frequency
TF	Term Frequency
IDF	Inverse Document Frequency
POS	Part-Of-Speech
BM25	Best Match 25
LMRT	Lifelog Moment Retrieval Task
IR	Information Retrieval
GPS	Global Positioning System
SQL	Structured Query Language
UTC	Universal Time Coordinated
NES	Named Entety Recognition
TACC	Texas Advanced Computer Center
GPU	Graphic Processsing Unit

CAPÍTULO 1

Introdução

"Photography is a way of feeling, of touching, of loving.
What you have caught on film is captured forever...
It remembers little things, long after you have forgotten everything."
— Aaron Siskind

Aduzimos, na nossa génese, um lado emocional influenciador e influenciável, que repercute a forma como vivenciamos momentos da nossa vida[1]. As emoções mais intensas, sobretudo as que transportam sentimentos positivos incitam-nos, ocasionalmente, ao desejo de perpetuar o momento que as provoca. Criar um objecto visual, como uma fotografia, permite que um instante das nossas vidas fique eternamente registado. Além disso, as imagens podem gerar impulsos que nos relembram emoções de um episódio longínquo[2][3]. Fazêmo-lo há séculos. Desde a pintura, passando pela transformação da arte e culminando na contemporânea fotografia. Registar momentos de uma forma visual é algo que acompanhou toda a nossa evolução.

A fotografia e a imagem são hoje maioritariamente digitais. Os dispositivos que as captam são cada vez mais pequenos e transportados diariamente pela generalidade da população. O que significa que, na actualidade, qualquer individuo eterniza episódios da sua vida, facilmente e em qualquer instante.

Com a digitalização, a imagem deixou de ser apenas uma memória visual. Hoje é vista em áreas fulcrais como uma ferramenta indispensável. É comum ver-se em engenharias ou ramos medicinais o aproveitamento dos benefícios fornecidos pelo processamento de imagem[4]. A análise do comportamento social[5] e a protecção dos cidadãos são alguns exemplos. Também novas tecnologias resultaram da aplicação da imagem, como a navegação assistida[6] ou reconhecimento de localização e ambiente envolvente[7][8].

Nestes pilares foram surgindo, ao longo dos anos, conceitos como *Egocentric vision*, *First-Person vision* ou *lifelogging*[4]. São abordagens, baseadas em visão por computador, cujos alicerces centram-se em analisar o comportamento social e melhorar a qualidade de vida. Para

tal, é necessário o registo gradual de actividades quotidianas de um individuo. *Lifelogging* surge precisamente neste contexto.

O *lifelogger* recorre a dispositivos sensoriais que lhe permitem registar detalhadamente os eventos do seu dia. Seja um *smartphone*, *lifelog cam*, *smart glasses* ou semelhantes, registam, para além das imagens, outro tipo de dados. No final, resulta um arquivo que pode corresponder a largos períodos de tempo e é composto por dados biométricos, temporais, localização e imagens. Consequentemente, os *datasets* ou *lifelogs*, como são apelidados, contam com um elevado nível de detalhe, mas também são bastante volumosos.

A recuperação de um especifico momento, nestes arquivos, só é viável recorrendo a sistemas computacionais. Nesse âmbito, têm surgido sistemas conhecidos por *retrieval systems (RS)* ou sistemas de recuperação de informação. As suas arquitecturas são desenhadas para analisarem um cenário detalhado pelo utilizador e devolverem o momento que o mesmo descreveu. Estes sistemas transformam as imagens arquivadas em elementos textuais que as descrevem [9][10][11]. Para tal, têm na sua estrutura algoritmos de análise e processamento de imagem e texto. A comparação entre a descrição registada pelo utilizador e os dados descritivos de cada momento, obriga a que os *retrieval systems* compreendam as ligações semânticas entre ambas as partes. Obrigatoriamente, é necessário educar o sistema computacional. Habitualmente, recorrem-se a algoritmos de análise textual baseados em *Natural Language Processing* [12]. *NLP* é um conceito coberto por várias técnicas e ferramentas que permitem reduzir distâncias entre a linguagem computacional e a linguagem do utilizador.

1.1 Motivação

- ***Volume e diversidade de dados***

  A portabilidade e a simplicidade que foram adicionadas, ao longo dos últimos anos, a dispositivos como câmaras, *smartphones* ou *smartwatches*, gerou um crescimento abrupto na utilização dos mesmos. Hoje em dia é frequente registar-se um passeio de bicicleta com uma câmara suportada no capacete ou contabilizar-se o número de passos no percurso para o emprego com recurso ao *smartwatch*. Além disso, o *smartphone* transformou-se num monitor pessoal de cada individuo. As múltiplas funções destes dispositivos permitem registar localizações, monitorizar períodos de sono ou registar episódios em imagem. Qualquer individuo produz dados diariamente. Portanto, a utilidade que estes dispositivos oferecem, repercute-se num número de informação extremamente amplo.

  Este problema é verificado em *lifelogs*. Para além de registarem períodos extensos do quotidiano do *lifelogger*, são minuciosamente detalhados por uma panóplia de dados. O gráfico da Figura 1.1 ilustra exactamente esse fenómeno. Retirado do estudo [13] demonstra o crescimento exponencial dos dados produzidos por um *lifelogger* durante um período de 10 anos. É ainda evidente o impacto do crescimento e evolução dos dispositivos de captura. Os anos mais recentes registam uma maior diversidade de dados, obrigando paralelamente ao alargamento da capacidade de armazenamento. Reforçando a questão, a Tabela 1.1 [14] mostra-nos a variedade e quantidade de dados que podem ser

arquivados em *lifelogs*. Dependendo do período de arquivo, um *dataset* poderá atingir as dezenas de *terabytes*.

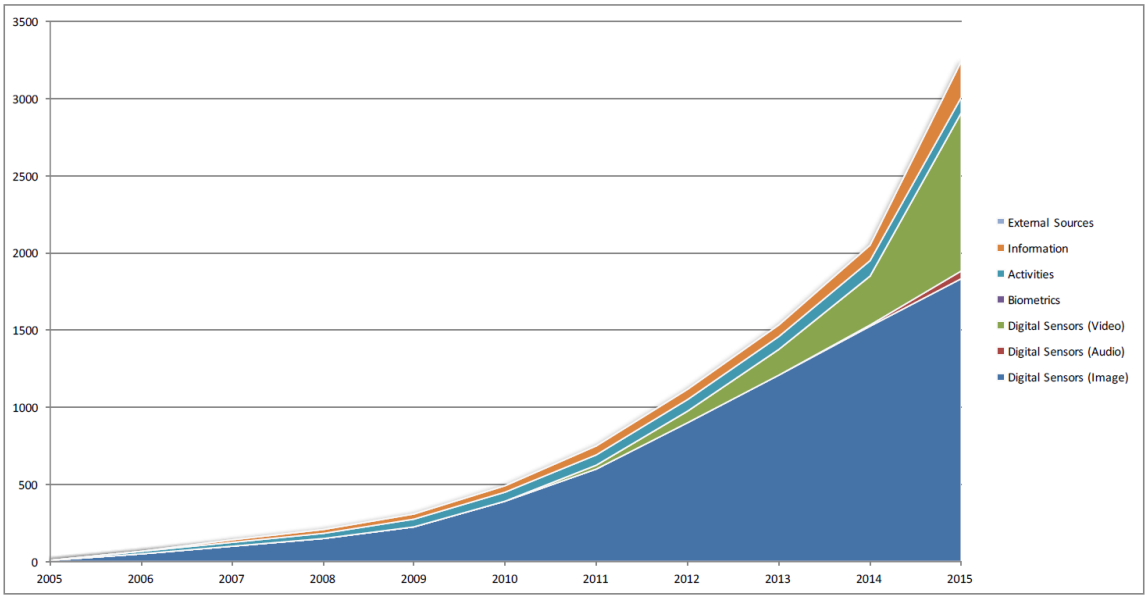


Figura 1.1: Quantidade de dados gerada por um *lifelogger* ao longo de 10 anos. [13]
Crescimento do armazenamento (eixo vertical (GB)) de 5GB de imagens e actividades (2005) para mais de 3200GB(3.2TB) de informação multi-modal como imagens, áudio, vídeo ou dados biométricos(2015).

Tabela 1.1: Ilustração da diversidade e quantidade de dados de *Lifelog*[14].

Content type	**Volume/day**	**In one year**	**In a life-time**
HD Video	5,840 hours	32.8TB	2.65PB
Autographer Camera	1.1 million images	479.6GB	40.8TB
Audio (mono - 22 KHz)	5,840 hours audio	227.8GB	19.4TB
Microsoft SenseCam	1.65 million images	30.2GB	2.6TB
Accelerometer (1 Hz)	21 million readings at 1 Hz	0.05GB	4.25GB
Locations (0.2 Hz)	3.9 million GPS points	0.01GB	1TB
Bluetooth Interactions	150,000 (estimated) encounters	2GB+	150GB
Reading Log	User dependent	1GB+	80GB

A questão não se prende essencialmente nas condições de armazenamento, mas sim na gestão dos dados. Se é verdade que um *lifelog* é extremamente extenso, também é verdade que é repleto de diferentes tipos de informação.

- ***Recuperação e classificação de momentos de um lifelog***

  O desafio que enfrentam os *retrieval systems* é precisamente recuperar um momento armazenado num arquivo tão amplo. Têm surgido arquitecturas que apresentam técnicas distintas para ultrapassar o problema. Para identificarem e recuperarem um momento estes sistemas têm de oferecer um "bom mecanismo de busca, associado a uma interface intuitiva", que resulte numa "pesquisa rápida e eficaz"[9]. Ora, um bom mecanismo tem de ser capaz de compreender o que o utilizador lhe comunica em linguagem natural.

  As técnicas de *NLP* são diversas e cruciais neste processo. Como relata a literatura, "*Natural Language Processing* relaciona-se com diferentes teorias e técnicas que lidam com o problema da linguagem natural para comunicar com computadores."[12]. O auxilio que presta na identificação automática de dados e no reconhecimento de correspondências entre sistema e utilizador, aumenta a precisão e eficiência nos resultados.

  A comunidade de investigação tem assumido um papel fundamental, contribuindo com inúmeros estudos e desenvolvimentos no conceito de *lifelog* e *retrieval systems*[15][11][16][4]. Um dos pontos em maior foco, actualmente, incide na classificação de resultados obtidos pelo sistema. As arquitecturas contam com um algoritmo baseado em funções matemáticas que é responsável por classificar a similaridade entre a questão imposta pelo utilizador e os resultados.

  Para motivar a comunidade de investigação surgem várias competições como *ImageCLEF*[1] e *Lifelog Search Challenge*[2] que permitem avaliar em tempo-real as propostas desenvolvidas pelos participantes. Têm também um papel importante na evolução destes sistemas, uma vez que são sempre apresentados obstáculos que obrigam a inovar os mecanismos.

- ***Lifelogging* e qualidade de vida**

  Melhorar a qualidade de vida é um dos motivos da evolução dos *Retrieval systems*. Um assunto ao qual se tem dado particular atenção incide nas doenças relacionadas com perda de memória. "De acordo com a *World Health Organization*, em 2017, o número de pessoas com demência atingiu os 47 milhões e em 2030 serão 75 milhões" [17]. *"Recollecting", "reminiscing", "remembering", "reflecting" e "retrieving"*, conhecidos pelos *5 R* são segundo *Sellen e Whitakker* [18] as cinco funções de memória que um *lifelog* pode suportar e oferecer. São estes os pilares que têm sido base para a aplicação destes sistemas na tentativa de melhorar ou reduzir o impacto de doenças como o *Alzheimer*. Podem contribuir para a simples identificação e localização de um objecto ou relembrar um momento que a doença fez esquecer. São elementos que se podem retirar da aplicação do conceito de *lifelog* e reduzir o impacto da doença no quotidiano.

Todos estes factores impulsionaram o interesse no estudo e desenvolvimento no contexto de *lifelogging*. O facto de se enfrentar a questão do crescimento da diversidade e quantidade de dados e o consequente desenvolvimento de sistemas que visam reduzir esse mesmo problema

[1] https://www.imageclef.org/
[2] http://lsc.dcu.ie/

conduziram à análise de elementos que necessitam de melhorias. A percepção do funcionamento destes sistemas alertou para a necessidade de resolução de lacunas na comunicação entre o utilizador e o mecanismo, bem como na classificação dos resultados devolvidos. Portanto, o desenvolvimento de um algoritmo baseado em técnicas de *NLP* e funções para a classificação de resultados surge com o intuito de resolver os problemas mencionados. Se a aplicação dos mesmos melhorar a *performance* de alguns sistemas isso significa que as áreas abrangidas pelo conceito sairão beneficiadas.

1.2 Contribuição

O trabalho realizado nesta dissertação pretende sinalizar e colmatar processos complexos na recuperação de momentos. A investigação e estudo efectuados no âmbito deste trabalho conduziram ao desenvolvimento de um algoritmo de processamento de texto e classificação de resultados. Este mecanismo foi projectado e desenvolvido com foco na possível adaptabilidade do mesmo a um sistema de recuperação de informação.

A elaboração deste documento retrata um levantamento geral do estado de arte com conceitos essenciais como *Lifelogging*, *Natural Language Processing* e classificação de resultados. A exploração destes temas objectiva demonstrar metodologias e ferramentas aplicáveis neste conceito. Além disso, serviu de base para a realização do algoritmo que permite estabelecer ligações comunicacionais, entre utilizador e sistema, com recurso a uma expansão de dicionário diversificada. Ademais, são introduzidas técnicas de reutilização de dados transformando-os em elementos essenciais na geração de crivos de informação.

A Figura 1.2 representa a base da arquitectura desenvolvida. O *query* do utilizador e os dados armazenados na base de dados são os elementos que constituem a entrada do sistema. Como referido na Secção 1.1 os dados advêm sobretudo de imagens, registos de localização, tempo, informação biométrica e descrição de actividades. No bloco *INPUT* os dados apresentam-se maioritariamente em formato textual. O sistema recebe termos que descrevem detalhadamente os momentos armazenados.

Destacam-se os estágios *NLP* e *RANKING*, uma vez que são os que receberam maiores contribuições. No bloco de *NLP* foram introduzidas as técnicas de expansão de dicionário e introduzidas ferramentas para o tratamento dos dados textuais. Neste bloco procedeu-se ainda a um reaproveitamento de dados transformando-os em conceitos temporais e de localização. Uma base de classificação de resultados combinando as funções *TFIDF* com similaridade de cosseno e BM25 foi o trabalho realizado no bloco *RANKING*.

A saída do sistema, representada pelo bloco *OUTPUT*, consiste na lista de melhores resultados com base na classificação do bloco *RANKING*. A lista é constituída por a identificação das imagens e respectivo *score* permitindo ao utilizador identificar as imagens que o sistema associou ao *query*.

A auxiliar o desenvolvimento e os blocos de testes do algoritmo esteve uma tecnologia *HPC*. Esta contribuição resulta de uma colaboração do *Texas Advanced Computer Center*[3] sediada

[3] https://www.tacc.utexas.edu/

na *University of Texas at Austin* com o Instituto de Engenharia Electrónica e Telemática de Aveiro (IEETA-UA).

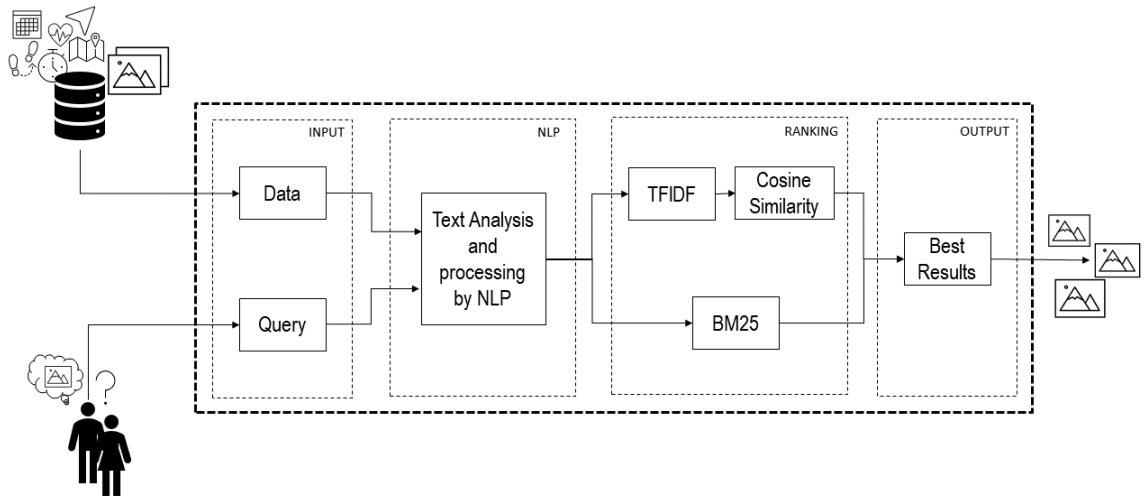


Figura 1.2: Estrutura base do algoritmo proposto na dissertação.

1.3 Estrutura do Documento

A matriz desta dissertação é dividida em 6 capítulos. No **Capítulo 2** é descrito o conceito de *Lifelogging*, bem como abordagens e metodologias aplicadas na recuperação de momentos em *datasets* com ampla diversidade de dados. No **Capítulo 3** é abordado o conceito de *Natural Language Processing* e são exploradas ferramentas essenciais para o processamento e análise de texto. Metodologias de classificação de resultados são também uma temática abordada neste Capítulo. No **Capítulo 4** é apresentada a proposta do algoritmo para a identificação e recuperação de momentos. Neste Capítulo, são detalhadas as intervenções aplicadas no mecanismo como técnicas de filtragem, análise e classificação de dados. O **Capítulo 5** descreve o comportamento e a *performance* do algoritmo com base em várias iterações e testes. Por fim, o **Capítulo 6** oferece uma síntese de todo o trabalho desenvolvido e uma visão geral de melhorias e direcções que o tema em estudo pode rumar.

CAPÍTULO 2

Lifelog Moment Retrieval

Registar as actividades do quotidiano, com recurso a sensores digitais, e armazenar permanentemente os múltiplos e diferenciados dados recolhidos é a definição mais consensual de *lifelogging*. Há ainda a ideia de que a actividade de *lifelogging* se inicia após a recolha de dados, ou seja, na fase de utilização dos mesmos[19]. No entanto, segundo a literatura[14], não há uma definição exacta de *lifelogging*. É ainda recorrente ver-se actividades que englobam memórias digitais pessoais, serem associadas ao conceito. *Lifeblog* ou *lifetime store* são alguns exemplos[14]. Um *lifelogger*, recorre a um ou vários dispositivos incorporados com sensores digitais, normalmente suportados no seu corpo e regista o seu ambiente envolvente em primeira pessoa[20](Figura 2.1).

O registo individual das actividades diárias, com um alto nível de detalhe, oferece ferramentas que beneficiam várias áreas. O facto de ser uma *"black box"*[21] com informações de como existimos e agimos é uma mais valia em ramos como medicina[16], saúde[17], psicologia[22] ou ciências. Melhorar a qualidade de vida é uma das vertentes com maior ênfase neste conceito, desde recuperar um simples episódio, passando pela análise ao comportamento social, finalizando no foco em doenças e problemas de memória.

O arquivo de toda a panóplia de dados recolhidos denomina-se *dataset* ou *lifelog*. O *lifelog* é um dos principais pontos de atenção, para as comunidades de investigação [18]. Pode reportar um longo período da vida de um individuo, ou pelo menos largos ciclos segmentados. Consequentemente, resulta uma enorme quantidade de dados para armazenamento. Este factor é um verdadeiro desafio na pesquisa e recuperação de momentos[23][21].

A recuperação de informação ou de um momento exige uma eficiente organização dos dados recolhidos bem como uma proficiente utilização dos mesmos. Esses são os primeiros passos para reduzir as distâncias comunicacionais entre o humano, que pretende recuperar um momento, e o sistema computacional, que tem em sua posse toda a informação.

(a) (b) (c) (d)

(a) a)GoPro b)SenseCam c)Looxcie d)Narrative Clip[20].

(b) a) Nike 'Fuelband', b) Contour wearable video camera with GPS, c) Vicon Revue wearable camera with sensors, d) Heart Rate Monitor display watch, e) Autographer from Vicon, f) GPS tracker, g) Jawboneï£¡s 'Up' Fitness Tracking Bracelet, h) Smartphone with built-in sensors, i) Audio recorder, j) fitbit 'One' Wireless Activity Tracker, k) fitbit 'Zip' Wireless Activity Tracker[21].

Figura 2.1: Câmaras e outros dispositivos sensoriais utilizados para a prática da actividade de *lifelogging*.

2.1 Recuperação de Informação

Como identificado, anteriormente, a detecção de um momento especifico numa colecção de dados, dissemelhantes, é o grande objectivo. *Lifelog moment retrieval (LMRT)* ou *lifelog retrieval*, são as denominações atribuídas a este conceito emergente. O desígnio é o desenvolvimento de arquitecturas ou sistemas que sejam capazes de recuperar momentos registados em imagem. Para que o automatismo resulte são necessárias duas temáticas essenciais: o computador tem de ser capaz de interpretar imagens e texto. Pode adicionar-se um terceiro elemento que é a capacidade do algoritmo relacionar ambos.

2.1.1 Estado de Arte

Steve Mann[24], um dos pioneiros no tema, começou por desenvolver pequenos dispositivos electrónicos, portáteis, com sensores incorporados e de baixo consumo de alimentação para capturar e criar *lifelogs*. As investigações desenvolvidas inicialmente eram imersas no desenvolvimento de ferramentas tecnológicas que permitissem capturar um maior nível de detalhe para um *lifelog*[25].

Gordon Bell[26] desenvolve aquele que é considerado o primeiro sistema de recuperação de momentos registados por um individuo. *MyLifeBits* é caracterizado por actuar como um "arquivo de vida" contava com uma arquitectura que em parte se transporta para os dias de hoje. O sistema fornecia uma pesquisa completamente textual. Além disso, já contava com anotações de texto e áudio. O modelo dispunha de uma organização baseada em *clustering*, o que facilitava a detecção e visualização dos momentos em pesquisa.

Cedo se identificou potencialidades na captura e identificação de actividades sociais e diárias[27]. O *smartphone* veio potencializar a actividade de *lifelogging* fornecendo a cada individuo um conjunto de recursos que permitem registar, esmiuçadamente, o seu quotidiano. Nesta perspectiva, exploraram-se várias aplicações e a influência do *smartphone*, como potencial engenho de *lifelogging*[28]. Toda a informação adquirida foi e é crucial para o desenvolvimento de arquitecturas e modelos de recuperação de informação.

2.1.2 Abordagens e métodos

As arquitecturas no âmbito de *LMRT* são caracterizadas pelos métodos de introdução de uma pesquisa, também denominada *query*. São sobretudo divididas em dois grandes grupos: *content-based* e *text-based*.

- ***Content-based* e *Text-based***

  Os sistemas de recuperação de informação ou *Retrieval systems* que pertencem ao grupo *content-based* recebem uma imagem semelhante ao momento que se tenciona recuperar: *query-by-example*[10][15]. O sistema analisa o *dataset* e devolve as imagens cujo conteúdo visual se assemelha ao do utilizador. O segundo grupo de sistemas analisam *queries* de texto e devolvem os resultados ou momentos cujas ligações semânticas coincidem.

  As abordagens relativas aos sistemas *text-based* são mais comuns, embora cada sistema tenha um mecanismo de acordo com a utilidade que pretende atribuir aos dados arquivados. Há ainda modelos que conjugam métodos distintos de introdução de *queries*.

Novas abordagens têm vindo a ser introduzidas e testadas. Porém, apesar de inovadoras, não podem ser desagregadas na totalidade das pesquisas por texto.

- ***Query-by-Sketch***

  Query-by-sketch, ou pesquisa por desenho[29][11], representa uma das novas abordagens aplicadas neste tipo de sistemas. O utilizador tem a possibilidade de desenhar, escolher uma cor ou uma forma geométrica. Este método permite, em alguns contextos, uma pesquisa mais rápida. Imagine-se um cenário de competição como os descritos na Secção 2.1.5:

  – Exemplo: *"I was playing football with a white ball."*

  O utilizador pode, de imediato, seleccionar ou desenhar um circulo, representando a bola *("ball")* e definir a cor branca *("white")* num selector de cores. Apenas terá de introduzir textualmente a actividade: *"playing football"*. Embora estas interacções ofereçam novas abordagens, na realidade as cores e as formas geométricas introduzidas são, também elas, processadas pelo sistema como texto.

- ***Gesture-based query***

  Mais recentemente, o mundo virtual tem encontrado lugar em alguns modelos[30]. Conhecida por *gesture-based query*, esta é uma metodologia que oferece ao utilizador uma interacção na qual pode seleccionar filtros ou conceitos que descrevem o momento que o mesmo pretende encontrar. O utilizador tem uma imersão no sistema sendo-lhe facultada uma interface 3D, na qual pode navegar com recurso a controladores.

- ***Query-by-voice***

  Também a pesquisa por voz foi introduzida em modelos cujo *software* faz a tradução para texto[31]. O utilizador questiona o sistema oralmente, no entanto, uma vez mais, a pesquisa é processada como texto.

- ***Interactive learning***

  Outro método importante e interessante trata-se de *interactive learning*. Os sistemas que recorrem a esta metodologia, têm em consideração o *feedback*, positivo ou negativo, do utilizador[32]. É apresentado ao utilizador um conjunto de imagens e o mesmo indica ao sistema quais as imagens que se identificam ou não com o momento preterido.

2.1.3 Arquitecturas

As arquitecturas dos *retrieval systems* têm vários elementos em comum. São módulos fundamentais na resposta aos pontos fulcrais para almejar o momento que se pretende encontrar. Maioritariamente, o modelo base é constituído por uma base de dados, uma interface e o elemento principal: motor de busca ou *search engine* (ver Figura 2.2).

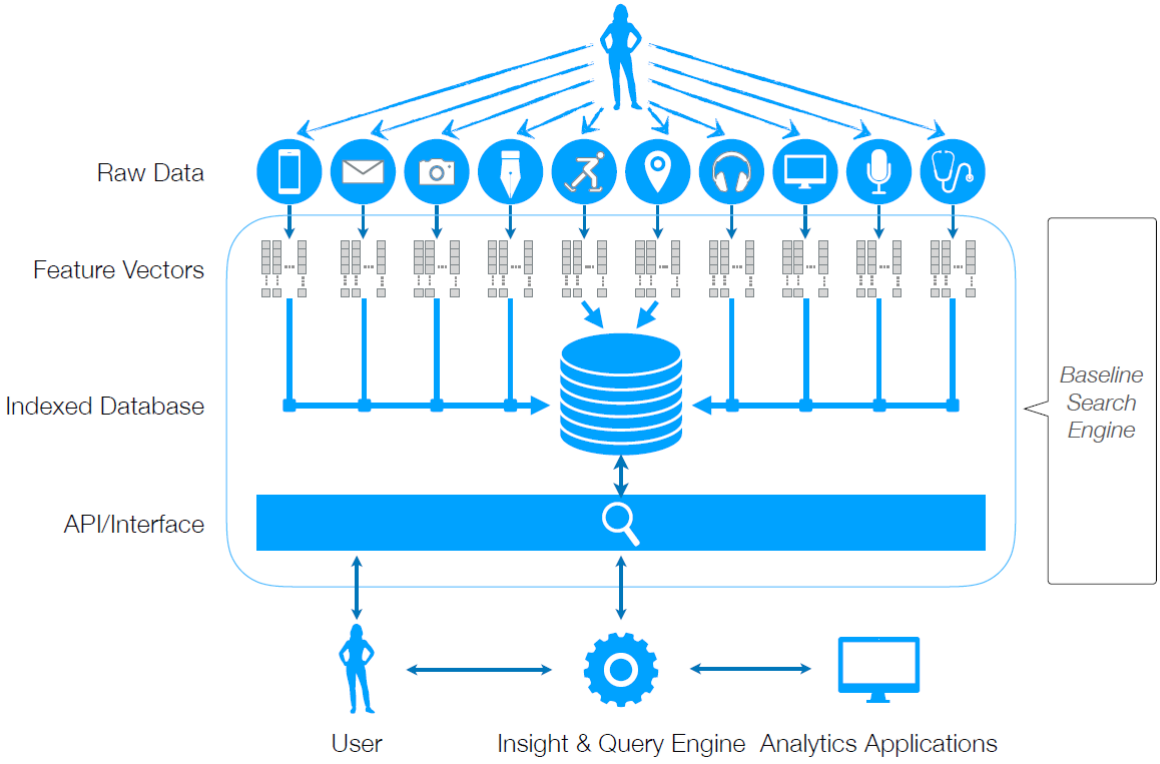


Figura 2.2: Modelo de arquitectura de um *Retrieval system*[23].

Um dos segredos para um sistema eficiente é uma base de dados bastante detalhada e organizada. Para isso, são aplicadas técnicas e algoritmos na organização e expansão dos dados do *lifelog (Secção 2.1.4)*.

A interface tem também um papel essencial, uma vez que é a ferramenta de interacção entre o utilizador e o sistema. O motor de busca é o responsável pela ligação das entradas registadas na interface, com os dados armazenados no banco de dados. "Um bom mecanismo de busca, associado a uma interface intuitiva são as componentes chave para um sistema competitivo."[33]. É essencial que o motor de busca interligue o conteúdo inserido no *query* do utilizador, habitualmente textual, com os dados que descrevem a imagem. Tal obriga a que se ensine o sistema a entender e perceber o conteúdo. Portanto, o motor de busca é constituído por algoritmos em três campos fundamentais: expansão dos elementos descritivos de cada imagem, análise e processamento de texto e classificação ou *ranking* de resultados.

2.1.4 Funcionamento

Predominantemente, os sistemas têm dois momentos: *offline* e *online*.

- ***Offline***

  No nível *offline* é efectuado o processamento dos dados ou *metadata*. O propósito é dilatar a informação detalhando e enriquecendo a base de dados. Deste modo, a probabilidade de *matching* entre a pesquisa do utilizador e o conteúdo do *lifelog* aumentará.

  A expansão de dados resulta da extracção de conteúdos e elementos das imagens e texto.

  – **Extracção de informação a partir de imagens**

  As arquitecturas aplicam algoritmos que permitem identificar nas imagens elementos como: pessoas, cores, objectos ou locais. Em suma, são extraídas as características das imagens que permitirão identificar um evento. Os algoritmos utilizados intitulam-se de *feature extraction algorithms* ou *object detection algorithms*[34] e são baseados em métodos de *machine learning*[35] e *computer vision*[19].

  Machine learning consiste em educar, com recurso aos dados fornecidos, um computador na identificação e selecção de padrões automaticamente. No contexto dos sistemas de recuperação de informação recorrem-se a redes neurais convolucionais *(CNN's)*[36], que aplicadas na análise das imagens obtêm as *features* ou elementos descritivos das imagens.

  Esse processo culmina com o conceito de anotação ou *annotation*. Aqui, o sistema regista textualmente todas as características obtidas nas imagens com recurso aos algoritmos mencionados anteriormente. A Figura 2.3 exemplifica a identificação de objectos numa imagem, que podem ser verificados na Tabela 2.1. A titulo exemplificativo a identificação foi realizada na *google cloud vision* API[37].

  * Exemplo: *"I was on the street with my bicycle."*

Figura 2.3: Simulação de resultado ao *query "I was on the street with my bicycle."*.

Tabela 2.1: *Objects* e *labels* detectados na Figura 2.3. *Objects* identificam um elemento numa imagem (ex: *Bicycle*). *Labels* descrevem e apontam elementos associados a *objects* (ex: *Bicycle tire, Bicycle Handlebar*).

Objects	Score	Labels	Score
Person	88%	Bicycle	97%
Person	82%	Building	94%
Bicycle	71%	Bicycle Tire	92%
Bicycle	65%	Sky	92%
		Bicycle Handlebar	92%
		Vehicle	90%
		Infrastructure	90%
		Tree	88%
		Road Surface	83%
		Road	79%

- **Extracção de informação a partir de texto**

  Os dados textuais provenientes do processo de registo do *lifelogger*, bem como os que resultam do processo de *annotation*, são também alvo de processamento antes de serem armazenados. Este processo consiste na aplicação de algoritmos alicerçados em técnicas de *Natural Language Processing* [12]. O recurso a *NLP* oportuniza a que o vocabulário descritivo recolhido seja expandido em vertentes como sinónimos, hipónimos ou hiperónimos. Ademais, a capacidade de extrair informação relevante de texto resulta também daquilo que é a capacidade destes algoritmos reconhecerem e identificarem, por exemplo, locais e actividades.

  O conceito de *Natural Language processing* é analisado em maior detalhe no Capítulo 3.

- ***Online***

  No estágio *online* ocorre tudo aquilo que acontece em tempo-real. O sistema recebe a informação introduzida pelo utilizador. Essencialmente inserido em formato de texto. Mais uma vez, o *retrieval system* recorre a técnicas *NLP* para identificar e seleccionar conceitos no *query* inserido.

  O ponto chave ocorre quando o sistema interpreta a entrada do utilizador e analisa o conteúdo previamente processado e armazenado. Para este processo, o motor de busca recorre a algoritmos para interpretação de linguagem natural.

  O último passo, no estágio *online*, concentra-se na classificação e apresentação dos melhores resultados ao utilizador. A classificação é obtida por técnicas e funções matemáticas que conjuntamente com mecanismos *NLP* analisam a similaridade entre pergunta e resposta. Na Secção 3.4, são identificadas duas funções que exemplificam técnicas frequentemente aplicadas e novas abordagens.

2.1.5 Desafios e Competições

Os *workshops* e competições surgem com a finalidade de criar desafios que permitam testar e melhorar sistemas ou modelos desenvolvidos para recuperar informação.

Com maior ênfase em *lifelogs* e nos problemas de gestão e recuperação num largo número de dados, surgiram desafios como *ImageCLEF*[38], *NTCIR lifelog*[39], *Video Browser Showdown (VBS)*[40], ou *Lifelog Search Challenge (LSC)*[41].

O principal objectivo entre o leque de competições é desenvolver infraestruturas que proporcionem bases e condições à comunidade de investigação para avaliarem os seus sistemas. Além de criar obstáculos aos sistemas, afim de os avaliar, também é um ponto de discussão de novas ideias e técnicas no seio da comunidade.

No *ImageClef* são avaliados métodos aplicados na recuperação de informação baseados na combinação de elementos textuais e visuais. Também, mecanismos de anotação automática de imagens e conceitos na formulação de *queries* são examinados. Nomeadamente a diversidade de ferramentas interactivas ao dispor do utilizador.

No mesmo sentido, o *LSC* desafia ao desenvolvimento de um sistema que permita aos utilizadores encontrar imagens especificas, relacionadas com um momento. Este evento ocorre num período de tempo limitado. Além disso, o sistema deve ainda ser capaz de classificar os resultados obtidos e permitir uma fácil interacção a *experts* e novatos. Actualmente o *dataset* cobre 115 dias do quotidiano de um individuo, resultando em aproximadamente 200 mil imagens armazenadas. Com a intenção de simular o raciocínio de uma pessoa, são lançadas pistas a cada 30 segundos, sobre o momento que se pretende encontrar e recuperar.

Em suma, estes eventos têm permitido o surgimento de novas e diversas arquitecturas, trazendo uma mais valia para o conceito. Inclusive alguns sistemas foram inicialmente desenvolvidos para desafios como o *Video Browser Search*, num contexto de vídeo e são adaptados para outros desafios como o *Lifelog Search challenge* num contexto de recuperação de informação em imagem [29][11][42].

- ***Exemplo de desafio aplicado no Lifelog Search Challenge***

Tabela 2.2: Conjunto de *queries* lançados em contexto de competição no *LSC*[11].

"I'm taking a photo of a white building with a unique blade-like design."
"The weather is cloudy and it is getting dark, being evening time."
"There are a number of buildings clearly visible in the image, including a hotel and a Norwegian style house."
"I had just walked from a sushi restaurant to the hotel where I was staying and I had taken the photo just before entering the hotel. A large yellow pipe is also visible in the image."
"Just before taking the photo, I had been walking beside the sea."
"This happened on a Wednesday."

Figura 2.4: Resultado final para o conjunto de *queries* da Tabela 2.2 [11].

2.2 *Datasets*

Só é possível extrair benefícios da aplicação de um *retrieval system* se o mesmo for aplicado num arquivo de dados abundante num detalhe específico a uma temática. Em grande parte, as aplicações pretendem analisar actividades quotidianas de um individuo e retirar informações que permitam a intervenção em áreas abrangidas pelo conceito. Portanto, o que se deseja é um *lifelog* capaz de narrar a vida de um *lifelogger*. Há quatro grandes princípios documentados na realização de um *dataset*[13]:

- ***Continuidade***

  Imagine-se que é pretendido analisar a actividade diária de um individuo na prática de um desporto. A análise deste comportamento requer vários elementos e registos frequentemente anotados. Ou seja, a continuidade no registo de eventos por parte do *lifelogger* é fundamental para que se possam comparar comportamentos e retirar as ilações requeridas nessa análise.
- ***Diversidade***

  A diversidade ou robustez refere-se á oferta de dados que o constituem. Conforme descrito na literatura: "Os *lifelogs* devem conter, pelo menos, quatro tipos de informação básicos"[13][21][43]:
  - Informação visual;
  - Dados biométricos;
  - Actividade;
  - Localização e tempo;
- ***Anonimato e Protecção***

  O nível de detalhe de um *dataset* pode ser invasivo para o *lifelogger* e expor pessoas que ficam registadas em imagens. É essencial garantir o anonimato. Os dados são, por isso, minuciosamente verificados e são-lhes removidos elementos como: caras, matriculas de veículos ou endereços e moradas. Além do mais, o arquivo de dados é protegido por *passwords* e só é disponibilizado com as devidas permissões.

2.2.1 Tipos de dados

Como mencionado anteriormente os tipos de dados que constituem o *dataset* são: imagens, dados biométricos, actividade e localização. A Figura 2.5 representa uma imagem do *dataset ImageCLEF* e na Figura 2.6 é possível verificar as diferentes tipologias de dados associadas a esse momento.

Figura 2.5: Imagem "*b*00000001_21*i*6*q*_20150223_070808*e.jpg*" do *dataset* ImageCLEF.


```
"b00000001_21i6bq_20150223_070808e.jpg": {
    "minute_id": "20150223_0708",
    "utc_time": "UTC_2015-02-23_07:08",
    "attributes": [
        "no horizon",
        "enclosed area",
        "man-made",
        "glass",
        "indoor lighting",
        "wood",
        "glossy",
        "natural light",
        "matte",
        "cloth"
    ],
    "categories": {
        "wet_bar": 0.057,
        "alcove": 0.046,
        "church/indoor": 0.045,
        "utility_room": 0.042,
        "shower": 0.037
    },
    "concepts": {
        "bottle": {
            "score": 0.987764418,
            "box": [
                595.58203125,
                448.6576874788142,
                618.7140239197531,
                511.7453828921988
            ]
        },
        "sink": {
            "score": 0.777268946,
            "box": [
                856.0205439814815,
                511.2142625919058,
                1014.1625192901234,
                568.4875147795874
            ]
        }
    },
    "local_time": "2015-02-23_07:08",
    "timezone": "Europe/Dublin",
    "longitude": -6.15827,
    "activity": "NULL",
    "location": "Home",
    "elevation": "NULL",
    "speed": "NULL",
    "heart": "NULL",
    "calories": "1.2062000036239624",
    "steps": "NULL"
```


Figura 2.6: Excerto de um ficheiro JSON no qual é registado detalhadamente um momento. Há registo de elementos temporais *(minute id, utc time)*, dados biométricos *(heart, calories)*, localização *(timezone, longitude)* e elementos descritivos como atributos, conceitos ou categorias.

- ***Dados visuais***

  As imagens são o elemento principal. Quando queremos recuperar um momento é precisamente uma imagem que queremos ver retornada pelo sistema. Ademais, são ainda uma fonte de novos dados informativos. Como analisado na Secção 2.1.4, são aplicados algoritmos de detecção e extracção de *features* nas imagens que vão enriquecer posteriormente a base de dados com recurso aos elementos detectados em cada imagem. Com recurso às anotações obtidas, vários sistemas optam por segmentar e endereçar as imagens com base nos seu atributos e conceitos semânticos[44][45].

  Em algumas situações opta-se por dividir em *clusters* cuja identificação é o elemento identificado com maior *score*. Se um algoritmo detectar um carro numa imagem e o seu *score* é mais elevado que os restantes elementos então a imagem será arquivada no *cluster: car* . Também é comum situações em que o contexto semântico entre objectos detectados é determinante para a segmentação dos dados. Note-se a situação apresentada no exemplo da Figura 2.3, na Secção 2.1.4, em que a imagem contém, entre outros, os elementos: *Bicycle - 97%, Bicycle Tire - 92% , Bicycle Handlebar - 92%* . Um ponto de ligação entre estes elementos será *Bicycle*. Neste caso a imagem seria arquivada no *cluster: "Bicycle"*.

- ***Dados biométricos***

  Os dados biométricos são sobretudo registos de batimentos cardíacos e gastos calóricos. Este grupo de dados não é o mais utilizado. Numa aplicação para a área da saúde[16] ou psicologia[22], os dados biométricos podem ser de utilidade máxima. No contexto de recuperação de momentos, por si só, a proficiência destes dados passa essencialmente por identificar se num determinado momento o individuo estava ou não em movimento. Atente-se ao seguinte cenário: *"Find the moment when I was resting in a blue bedroom"*. Neste caso poderia-se aplicar um filtro nos dados biométricos, identificando momentos em que o batimento cardíaco é mais baixo. Assim, resumiria-se a busca aos momentos em que a probabilidade do individuo estar a descansar é mais elevada.

- ***Actividades***

  As actividades são dados provenientes de aplicações de *smartphone* ou *smartwatches*. Resultam da monitorização de actividades diárias e centralizam-se em número de passos, tempo de repouso ou distância percorrida.

  O número de passos pode ser aplicado para identificar se o individuo está em movimento ou estagnado. Já a distância permite o reconhecimento de momentos em que o *lifelogger* está em viagem. A identificação de padrões relativamente à distância percorrida permite identificar ciclos quotidianos. Uma caminhada, o caminho para o trabalho ou o regresso a casa são actividades geralmente rotineiras.

  Outra vantagem destes dados serem fornecidos por aplicações moveis, focadas em actividades físicas ou cuidados de saúde, é a identificação da exacta actividade física que se está a praticar[21].

- ***Localização e tempo***

  A localização é posteriormente às imagens o tipo de dados mais utilizado no sistemas de recuperação de informação. Enquanto que os restantes dados respondem a 'O quê?', 'Quem?' e 'Como?', os dados GPS, respondem à pergunta 'Onde?' e os registos temporais atendem à questão 'Quando?'.

  Esta parelha de dados é utilizada por *retrieval systems* para duas finalidades: filtragem e organização. Seja em formato de longitude e latitude ou *timezone*, os *datasets* oferecem a hipótese de criar uma filtragem geográfica ou organizar e segmentar as imagens por continentes, países ou regiões[15],[32].

  Os registos temporais são habitualmente definidos por hora e data. São elementos que permitem enriquecer o conteúdo em arquivo, uma vez que se pode transformar a hora em momentos do dia e a data, textualmente, em anos, meses e dias. Ora, à imagem da localização, são componentes que permitem a aplicação de filtros e a organização de imagens em *clusters*.

  O modelo *BIDAL-HCMUS*[10] aplica uma técnica de organização de dados que engloba várias vertentes. Inicialmente aplica uma segmentação de dados por localização e tempo. Posteriormente, divide as imagens em *clusters*, cujo conteúdo corresponde. Por exemplo dois momentos registados na Noruega *("Norway")* com elementos nas imagens que indiquem que se trata de uma cozinha *("kitchen")* são armazenados no *cluster: (Norway)* que inclui o *cluster (kitchen)*. Esta metodologia denomina-se *atomic clustering*.

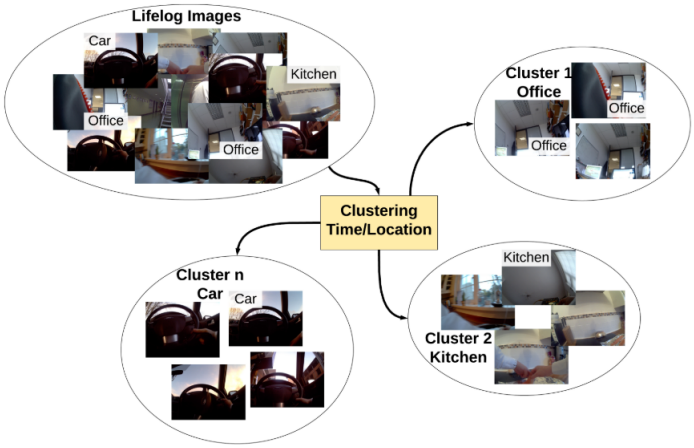


Figura 2.7: *Atomic clustering* baseado em localização e dados temporais [10].

CAPÍTULO 3

Natural Language Processing

"A linguagem é fascinante, pois pode ser um veículo para a comunicação de algo fascinante"[46]. Esta é uma das premissas presentes na base do surgimento de *Natural language processing*.

A grande barreira que *NLP* pretende quebrar é a distância presente na comunicação entre a linguagem natural ou humana e um sistema computacional. *Natural Language Processing* é um ramo de *Artificial Intelligence and Linguistics* focado em tornar mais fácil a interacção de um utilizador humano com um computador[12]. Esta distância comunicacional é reduzida, respeitando uma base ética[47], recorrendo a dois conceitos fundamentais: atribuir bases do conhecimento humano a sistemas computacionais e programá-los para recorrerem desse conhecimento no processo de compreensão [46]. Para tal, engloba um conjunto de metodologias e ferramentas para analisar, gerar, processar e manipular a linguagem natural [48].

As vantagens que *NLP* oferece despertam atenção não só a cientistas e engenheiros de *software* [49] [50], mas também em profissionais no ramo da linguística, psicologia e filosofia. As suas aplicações são distribuídas em campos como *Machine Translation*, *Email spam detection* [51], *Information Extraction* [52] [53], *Summarization* ou *Question Answering*.

3.1 Estrutura

Natural Language Processing é habitualmente dividida em duas partes: *Natural Language Understanding e Natural Language Generation* (Figura 3.1). Em suma, uma estrutura orientada a compreender e gerar linguagem natural.[12]

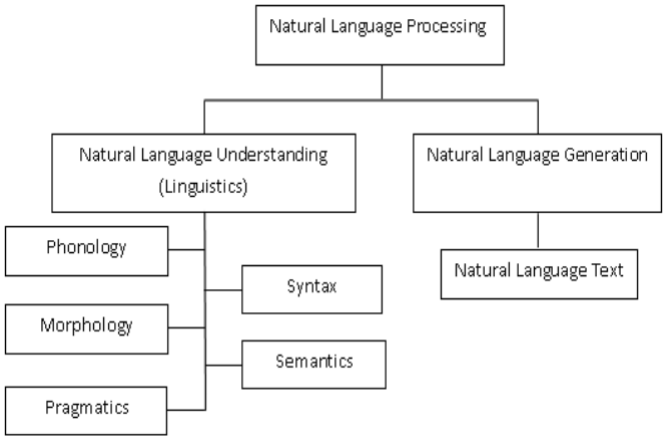


Figura 3.1: Estrutura NLP [12].

- ***Natural Language Generation***

  NLG é o processo de produção de palavras, frases ou parágrafos relevantes de uma representação interna. Segundo a literatura desenvolve-se em quatro etapas: identificação de objectivos, planeamento, avaliação das fontes de comunicação e realização textual dos objectivos.

- ***Natural Language Understanding***

  Neste processo ocorre a compreensão de palavras, frases ou parágrafos. Como se pode verificar na Figura 3.1, *NLU* aborda cinco terminologias importantes para a compreensão de linguagem natural.

  Fonologia (Phonology) - Disciplina da linguística que estuda e descreve os sons como unidades distintas (fonemas) e a sua função no sistema linguístico;

  Sintaxe (Syntax) - Parte da gramática que estuda e descreve as relações que as palavras estabelecem entre si numa frase;

  Morfologia (Morphology) - Disciplina da linguística que descreve e analisa a estrutura interna das palavras, bem como os processos de formação e variação de palavras

  Semântica (Semantics) - Disciplina da linguística focada no significado das expressões linguísticas (sejam elas fonemas, morfemas, palavras, sintagmas, frases) bem como das relações de significado que essas expressões estabelecem entre si.

  Pragmática (Pragmatics) - Disciplina que estuda as relações existentes entre as formas linguísticas e os falantes, no sentido de descrever o uso que estes fazem da língua nas mais diversas situações de comunicação

3.1.1 Problema: Ambiguidade

O principal problema que *NLP* enfrenta denomina-se ambiguidade. Este problema é habitualmente definido como o duplo sentido que uma frase ou palavra podem transportar. Esta adversidade pode manifestar-se ao nível lexical, sintáctico ou morfológico [54].

Ambiguidade lexical - Refere-se a palavras que podem ser classificadas em mais do que uma classe como nomes ou verbos.

- Ex: “duck” - nome(inglês) | “duck” - verbo(inglês)

Ambiguidade sintáctica ou estrutural - Sinaliza-se quando uma frase pode ter significados e sentidos opostos.

- Ex: “I saw the man with a telescope”.

O telescópio pode ter sido usado para observar o homem ou pode ter sido observado junto do homem.

Ambiguidade semântica - Quando uma palavra tem múltiplos significados.

- Ex: “go” - verbo(inglês) e com mais de 10 significados no vocabulário inglês.

Embora sejam conhecidas mais variantes como a ambiguidade pragmática ou referencial, por vezes verificam-se situações em que ocorrem em conjunto, agravando o problema. Para resolver a questão são aplicados métodos como *Minimising ambiguity*, *Preserving ambiguity*, *Interative desambiguity* e *Weithing Ambiguity* [55].

3.1.2 Conceitos

As áreas de aplicação e estudo no âmbito de *Natural language processing* são também descrições de técnicas e métodos aplicados no leque de campos abrangidos pelo conceito.

Tokenization - Primeiro processo aplicado em qualquer desenvolvimento *NLP*. O texto de entrada é segmentado em unidades linguísticas como palavras, pontuação, números ou alfanuméricos. Estas unidades denominam-se *"tokens"*. A existência de espaços entre as palavras facilita a segmentação[48].

Part of speech tagging (POS tagging) - Uma palavra pode ser classificada numa ou mais categorias gramaticais como nomes, verbos, adjectivos e artigos. *POS tag* funciona como uma simbologia capaz de identificar as diferentes categorias. É uma ferramenta bastante útil em *Machine Translation*, mas também bastante requisitada nas restantes aplicações [48] [56].

- Ex: "The ball is red- The/AT ball/NN is/VB red/JJ

Named entity recognition (NES) - Quando aplicada permite identificar e determinar palavras que se relacionam num texto com nomes próprios [12] [57].

CO-Reference Resolution - Processamento de frases ou textos para identificar palavras que se referem ao mesmo objecto.

Machine Translation - É um conjunto de métodos aplicados com o objectivo de traduzir automaticamente um texto para outro texto fluente, num outro idioma de linguagem natural, sem interferência humana. É uma das áreas mais complexas de *NLP* [48].

Morphologic segmentation - Consiste em segmentar palavras em morfemas, e identificar a classe dos mesmos.

Sentiment analysis - Conjunto de técnicas cujo objectivo é o reconhecimento de sentimentos presentes num texto [58]. As mesmas ferramentas são aplicadas para a detecção de emoções *(emotion detection)* [59].

Stemming - É um processo de *NLP* para a remoção de sufixos derivacionais ou sufixos que alteram a forma das palavras e a sua função gramatical. Após a eliminação dos sufixos restam os *stems*. É uma técnica muito aplicada em *retrieval systems* para garantir que as variantes de uma palavra não são deixadas de parte.

Tabela 3.1: Exemplo de alterações implementadas, com base em sufixos, pelo conceito *stemming*.

Original word	Stemming
program	program
programs	program
programmer	program
programming	program

Tabela 3.2: Exemplo de acções particulares de *stemming*.

Original word	Stemming
studies	studi
studying	study

É um recurso há muito requisitado, existindo já algumas ferramentas para a sua aplicação [60]. *Paice/Husk Stemmer, Lovin's Stemmer, Dawson Stemmer, Krovetz Stemmer, Xerox Stemmer, N-gram Stemmer, Snowball Stemmer, Lancaster Stemmer*, são alguns exemplos disponíveis [61]. O mais utilizado é o *Porter's Stemmer* [62] devido à sua rápida *performance* e simplicidade de aplicação. É no entanto limitado ao idioma inglês.

Lemmatization - é uma técnica que permite verificar as formas flexionadas de uma palavra para que possam ser analisadas como um único *lemma*, ou seja, a sua forma original. Tal como *Stemming* a sua aplicação deve-se à necessidade de garantir que todas as palavras num *query* são consideradas independentemente da forma em que se encontram.

Tabela 3.3: Exemplo da redução de um termo ao seu *lemma* com base no conceito *Lemmatization*.

Original word	Lemmatization
studies	study
studying	study

O *lemmatizer* mais aplicado é o *WordNet Lemmatizer* [63] . Trata-se de uma base de dados lexical que providencia relações semânticas entre palavras.

3.2 Recuperação de Informação

Como mencionado no inicio deste Capítulo, *Natural language processing* é aplicada em vários contextos. Em virtude desta dissertação, importa destacar o ramo de *information retrieval (IR)*. Este conceito é adoptado por sistemas de recuperação de informação *(retrieval systems)* explorados na Secção 2.1.

O campo de *information retrieval*, aliado a metodologias *NLP*, visa garantir a transmissão eficiente e efectiva de informação entre "o humano gerador de dados e o humano utilizador"[61]. Se transportarmos para o cenário de *lifelog*, os dados capturados e armazenados por um individuo são a informação que o mesmo ou outro individuo pretende obter.

Com a introdução de ferramentas para linguagem natural, neste processo, reduzem-se ambiguidades e distâncias de comunicação entre homem e sistema computacional[12]. Esta aliança de conceitos, "não só pretende encontrar a informação correta, mas também representar resultados de uma forma simplificada" [61].

Antes dos sistemas da Secção 2.1, ambos os conceitos eram aplicados em *query systems*. São a base dos sistemas estudados nesta dissertação. A sua arquitectura está representada na Figura 3.2[54]. As etapas retratadas, ainda que simplificadas, visam a compreensão semântica e a correcta interpretação da questão do utilizador, em linguagem natural e a devolução exacta do conteúdo.

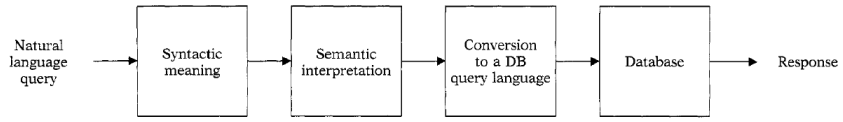


Figura 3.2: Arquitectura base de um *query system* [54].

3.2.1 Abordagens

Num contexto de recuperação de informação destinado a *lifelog* são primordiais técnicas mais complexas. A exigência de compreender as várias vertentes de texto presentes obrigam a que se considerem dicionários volumosos. Para tal, expandir o vocabulário de palavras que descrevem um momento, ou considerar todas as possíveis entradas de texto de um utilizador, independentemente da sua forma, são factores a considerar.

No leque de *NLP* as principais abordagens aplicadas nestes sistemas são baseadas em *machine learning* ou bibliotecas desenvolvidas e suportadas por sistemas lexicais. *Word2Vec*[64], *fastText*[65], *GloVe* são ferramentas no contexto de *word embeddings*[66] cujo objectivo é obter a informação semântica de palavras armazenadas em vectores. Cada palavra é armazenada em formato numérico num vector e as mesmas são organizadas tendo em consideração a ligação semântica (contexto e significado) que as relaciona. Este processo de mapeamento é realizado com recursos de *deep learning* e redes neurais. *Bag of words*[67] é uma metodologia idêntica, mas contrariamente a *word embeddings* o armazenamento é realizado em vectores de maior dimensão.

O recurso a bibliotecas desenvolvidas para aplicação de ferramentas *NLP* como *Stemming* e *Lemmatization* é também uma opção correntemente seleccionada. A expansão de dicionário ou vocabulário passa por explorar ligações entre palavras como hiperónimos, hipónimos e sinónimos. Sistemas lexicais como *WordNet*[68] são recursos preteridos na realização desta tarefa.

3.2.2 WordNet

WordNet é uma base de dados lexical no idioma inglês[69]. A sua organização consiste em agrupar nomes, verbos, adjectivos e advérbios em conjuntos de sinónimos cognitivos, intitulados: *synsets*. A ligação entre *synsets* é realizada na consideração de relações semânticas e lexicais.

WordNet não faz apenas estas interligações semânticas e lexicais como conhece e específica o sentido de cada palavra, o que a difere de um simples dicionário de sinónimos. Ademais, distingue substantivos comuns ou elementos como pessoas, países ou entidades geográficas mais específicas. Esta árvore de palavras que enriquece a base de dados, torna esta estrutura bastante requisitada [68].

A relação mais frequente aplicada entre *synsets* é a relação super subordinada ou hiperónima. Um hiperónimo é uma palavra cujo significado generalizado inclui o significado de outra palavra. A palavra *"flowers"*, na Figura 3.3 é hiperónima de *"rose"*, *"jasmine"* e *"orchid"*, que são denominações de flores *("flowers")*. Já um hipónimo é uma palavra cujo significado relaciona-a com outra palavra no qual o sentido é mais geral. Na Figura 3.3 pode verificar-se a ligação entre *"white"* e *"colour"*. A palavra *"white"* representa uma cor logo é um hipónimo de *("colour")*.

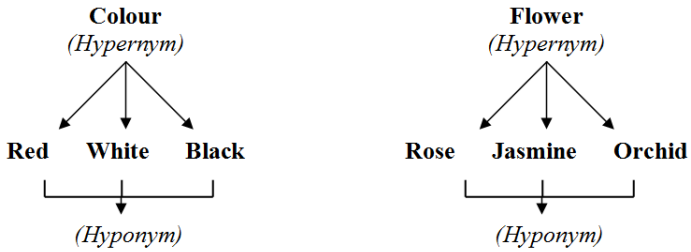


Figura 3.3: Hipónimos e Hiperónimos.

Os *synsets* são ainda interligados por sinónimos ou antónimos que englobam nomes, verbos ou adjectivos. Da Figura 3.4 [70] rapidamente se observa como uma simples palavra gera uma ramificação de outros termos. Quando o processo se repete para os termos gerados, desenvolve-se uma árvore de palavras com múltiplas ligações.

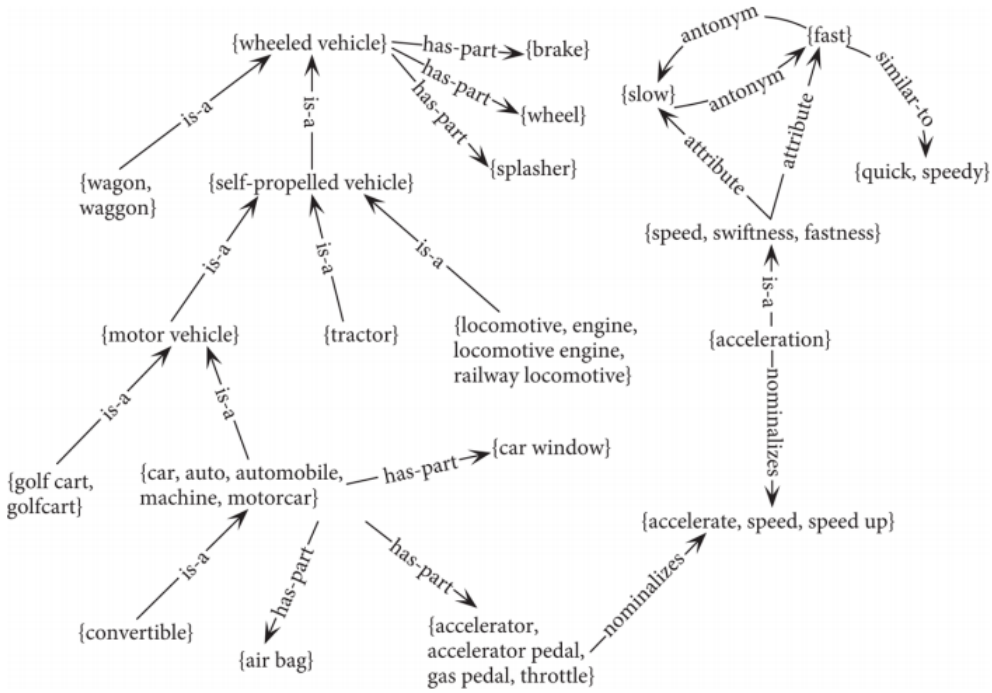


Figura 3.4: Relações entre palavras em *wordNet*[70].

3.3 Bibliotecas *Python*

A linguagem *Python* oferece um vasto leque de bibliotecas e ferramentas[71] para a aplicação das metodologias de processamento e análise de dados textuais com recurso aos conceitos *NLP* descritos anteriormente. Além disso, *Python* é uma linguagem interactiva, o que a torna particularmente útil no desenvolvimento de protótipos e projectos de engenharia de *software*[72].

NLTK - *Natural Language Toolkit* é uma das principais plataformas de apoio ao processamento de dados em linguagem natural[73]. Esta biblioteca é aplicada em contexto de investigação e desenvolvimento de metodologias *NLP*[74], pois oferece um conjunto de tutorias e módulos exemplares para diversas utilizações. *NLTK* facilita ainda a aplicação de *interfaces* como *WordNet*. Proporciona recursos a diversos conceitos como *text tokenization*, *sentence detection*, *lemmatization*, *stemming* ou *POS tagging*.

Gensim - Desenhada inicialmente para analisar a similaridade entre documentos, *Gensim* é uma biblioteca *open source* que suporta, entre outros, a modelização de vectores de palavras e tópicos[72][75][76]. *Topic modeling* é uma técnica de identificação e extracção de tópicos em textos e uma das principais ferramentas de *Gensim*. Na abordagem de *word embeddings*, algoritmos como *word2vec* e *fastText* podem ser implementados associados a esta biblioteca. Contrariamente a *NLTK*, a biblioteca *Gensim* pode ser aplicada na análise de longos *datasets* e é reforçada por bibliotecas como *SciPy* ou *NumPy*.

SpaCy - É uma biblioteca *open source* projectada para largos volumes de dados. Frequentemente aplicada no pré-processamento de texto para *deep learning*[77]. Esta fornece condições para o desenvolvimento de sistemas de compreensão de linguagem natural e sistemas de recuperação de informação. Esta biblioteca inclui ferramentas para *tokenization* em mais de 49 idiomas, *Named entity recognition*, *lemmatization* ou *POS tagging*.

coreNLP - *Stanford CoreNLP* é uma biblioteca estruturada em *Java* mas com vertentes para outras linguagens como *Python*. O objectivo dos desenvolvedores na *Stanford University*, era criar uma ferramenta eficaz e de fácil aplicação para a análise em linguagem natural[78]. *CoreNLP* oferece ferramentas para extrair várias propriedades de texto como *Named entity recognition*, *POS tagging* e ainda reconhecimento de padrões linguísticos, *pattern recognition*.

TextBlob - É uma biblioteca *Python* da qual se podem retirar ferramentas para processamento e analise de dados textuais. *TextBlob* é caracterizada por interfaces intuitivas e de fácil manipulação. A API fornecida permite aplicar um grupo de ferramentas *NLP* como *POS tagging*, *n-gram search*, *sentiment analysis*, *machine translation* ou *noun phrase extraction*[79]. Permite ainda a integração de recursos como *WordNet*.

PyNLPl - Pronúncia-se "pineapple". Esta biblioteca é aplicada em simples tarefas de *NLP* como a verificação da frequência com que ocorre um termo, ou identificação e extracção de tópicos[80].

Pattern - Construída para análise de conteúdos textuais na *web*, análises de redes e *machine learning* [81]. *Pattern* é por isso bastante valorizada para investigação e em desenvolvimentos *NLP*. Com recursos para *data mining (Google, Twitter, Wikipedia API)* a sua utilização é uma mais valia para *POS tagging* ou *sentiment analysis*.

Scikit-learn ou Sklearn - É uma biblioteca Python que oferece vários mecanismos de classificação, regressão e *clustering*. Os seus algoritmos permitem a aplicação de ferramentas como a função *TFIDF*, *k-means* ou *gradient boosting*. *Scikit-Learn*[82] é projectada para interagir com bibliotecas numéricas como *NumPy* e *SciPy*.

3.4 Classificação de Resultados

Neste ponto, tem-se de um lado a descrição detalhada de um momento, num *query*, e de outro lado um *lifelog* repleto de momentos descritos, em múltiplos termos. Pretende-se que os resultados mais relevantes, aqueles que são semelhantes à descrição do utilizador, sejam devolvidos. Para isso é iminente uma classificação de resultados.

É sobretudo uma questão de similaridade entre ambas as partes. Para tal, a atribuição de pesos aos vectores é fundamental para devolver a informação que corresponde às necessidades do utilizador.

A classificação de resultados tem sido alvo de estudos que visam melhorar a precisão das classificações com particular atenção na atribuição de pesos ou valores correctos, de forma a definir com critério os resultados finais[83].

Nesta dissertação é aplicada uma técnica bastante recorrente com recurso à função *TFIDF* associada a similaridade de cossenos. No entanto, sendo conhecidos os problemas desta metodologia, a mesma é reforçada com a introdução da função *BM25*.

3.4.1 TFIDF *Term Frequency Inverse Document Frequency*

TFIDF é uma função matemática de vectorização requisitada para suportar a classificação de resultados baseada na relevância atribuída a palavras presentes num documento ou vector e na influência dos mesmos num *dataset* . É um método frequentemente aplicado em *retrieval systems*[84][85], para diversas aplicações como análise de sentimentos *(sentiment classification)*[83] ou análise de texto *(text mining)*[63][86].

No seu *modus operandi* são distinguidas duas matrizes: *term frequency* e *inverse term frequency*.

3.4.2 *Term Frequency*

Term frequency ou frequência de um termo, classifica com base na ocorrência de um termo num documento ou vector. É representada pela Equação 3.1.

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{3.1}$$

O numerador corresponde ao número de ocorrências de um termo no documento d_j, enquanto que o denominador corresponde ao conjunto de ocorrências de todos os termos no documento d_j.

- Exemplo:
  query : [garden, trees]
  Vector 1 : [garden, trees, sunny, day, summer]
  Vector 2 : [houses, garden, backyard , houses]
  Vector 3 : [trees, birds, clouds, foliage, green, wood]

Tabela 3.4: Cálculo e valores de *Term frequency* para o vector 1, com base nas ocorrências de um termo e numero total de termos no vector.

vector 1	garden	trees	sunny	day	summer
ocorrências	1	1	1	1	1
TF	$1/5 = 0.2$	0.2	0.2	0.2	0.2

Tabela 3.5: Cálculo e valores de *Term frequency* para o vector 2, com base nas ocorrências de um termo e numero total de termos no vector.

vector 2	houses	garden	backyard
ocorrências	2	1	1
TF	$2/4 = 0.5$	0.25	0.25

Tabela 3.6: Cálculo e valores de *Term frequency* para o vector 3, com base nas ocorrências de um termo e numero total de termos no vector.

vector 3	trees	birds	clouds	foliage	green	wood
ocorrências	1	1	1	1	1	1
TF	$1/6 = 0.16$	0.16	0.16	0.16	0.16	0.16

Como se pode verificar nas Tabelas 3.4, 3.5 e 3.6, o cálculo de *TF* por si só não garante uma classificação aceitável para a recuperação de um momento. Se verificarmos, o resultado com melhor classificação seria o vector 2 uma vez que o termo *"houses"* apresenta um valor mais elevado. Esta classificação deve-se ao facto do termo *"houses"* aparecer duas vezes no vector 2, cujo comprimento é menor que qualquer um dos restantes em avaliação. O número de palavras presentes num vector bem como o número de ocorrências de um termo influenciam o resultado. Quanto menor for o vector mais relevância terá um termo, comparativamente a um vector de maior comprimento.

3.4.3 *Inverse Document Frequency*

Inverse document frequency ou frequência inversa de um documento, surge para rectificar as falhas de *TF*. Ou seja, favorece termos que aparecem com menor frequência nos documentos ou vectores de um *dataset*. É representada pela Equação 3.2.

$$IDF_i = log\frac{|D|}{|d_j : t_i \in d_j|} \tag{3.2}$$

O numerador corresponde ao total número de documentos, enquanto que o denominador corresponde ao número total de documentos nos quais o termo t_i está presente.

Tabela 3.7: Cálculo e resultados de *IDF* para cada termo do exemplo, considerando o número de vectores em que um termo surge e o número total de vectores

Termos	IDF
garden	$log(3/2) = 0.40$
trees	$log(3/2) = 0.40$
sunny	$log(3/1) = 1.09$
day	1.09
summer	1.09
houses	1.09
backyard	1.09
birds	1.09
clouds	1.09
wood	1.09
green	1.09

A Tabela 3.7 exemplifica o impacto do cálculo de *IDF* na classificação de resultados. Os termos com menor ocorrência em vectores são destacados relativamente aos que aparecem

em múltiplos vectores. Os termos *"garden"e "trees"* surgem em dois vectores e por isso é-lhes atribuído um valor mais baixo.

3.4.4 *TFIDF*

Um termo determinante deve ser aquele que distingue um vector ou documento da restante colecção ou *dataset*. Isto implica que os melhores termos tenham valores elevados de *TF*, mas ao mesmo tempo registem valores baixos relativamente ao restante *dataset*.

A função *TFIDF*, representada na Equação 3.4, tem precisamente esse papel. Determina a frequência relativa dos termos num vector especifico, comparando a proporção inversa de um termo perante todo o grupo de vectores. O cálculo resultante, intuitivamente, indica a relevância que um termo pode ter num vector.

$$TFIDF_{i,j} = TF_{i,j} * IDF_i \tag{3.3}$$

$$TFIDF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} * log \frac{|D|}{|d_j : t_i \in d_j|} \tag{3.4}$$

Tabela 3.8: Resultados *TFIDF* de cada termo com base no cálculo de *TF* e *IDF*.

Termos	vector 1	vector 2	vector 3
garden	0.08	0.1	0
trees	0.08	0	0.064
sunny	0.218	0	0
day	0.218	0	0
summer	0.218	0	0
houses	0	0.545	0
backyard	0	0.272	0
birds	0	0	0.174
clouds	0	0	0.174
wood	0	0	0.174
green	0	0	0.174

- ***Cosine Similarity***

  O que se pretende é recuperar o vector que mais se identifica com o *query* do utilizador. Para tal, interliga-se a função *TFIDF* com a similaridade de cosseno.

  O seu funcionamento passa por comparar dois vectores (*query* e vector de palavras) num espaço vectorial, avaliando o ângulo entre eles (Equação 3.5). O ângulo no contexto de recuperação de informação simboliza o quão semelhantes são o *query* e documento ou vector. A gama de classificação desta função trigonométrica é [-1,1]. Significa que se coincidirem na totalidade o valor será 1.

$$cos(\theta) = \frac{query \cdot vector_i}{\| \, query \, \| \| \, vector_i \, \|} \tag{3.5}$$

Começando por calcular o *TFIDF* do *query* tem-se na Tabela 3.9:

Tabela 3.9: Cálculo de *TFIDF* para o *query* do exemplo em estudo

Termos	TF	IDF	TFIDF
garden	0.5	0.40	0.2
trees	0.5	0.40	0.2

Aplicando a similaridade de cosseno com a fórmula 3.5 obtêm-se os resultados da Tabela 3.10 representados no gráfico da Figura 3.5.

Tabela 3.10: Resultados do cálculo de similaridade de cosseno com base nos valores *TFIDF* do query e vectores.

Termos	vector 1	vector 2	vector 3
cosine	1	0.2	0.2

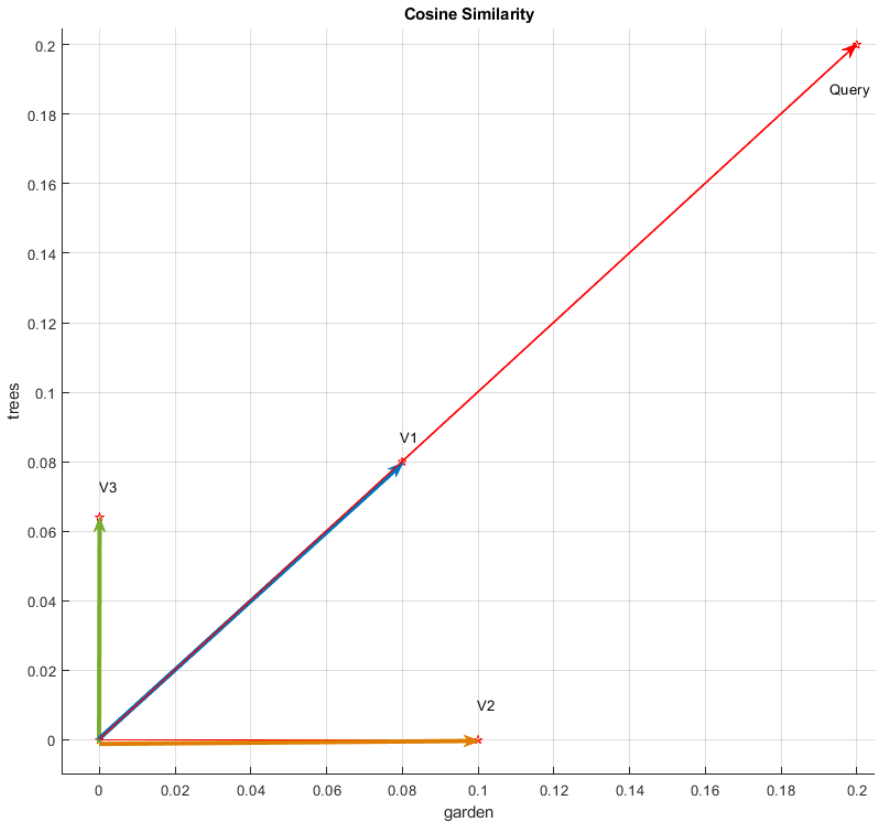


Figura 3.5: Espaço vectorial resultante da análise por similaridade de cosseno, no qual se verifica que o vector 1 é o que tem maior similaridade com o *query*. O ângulo do vector 1 é igual ao do vector *query*, enquanto que os restantes vectores tem um ângulo mais desfasado.

3.4.5 BM25

BM25 ou *Best Match 25*[87] é também uma função de classificação para recuperação de informação, que pode ser vista como uma melhoria à função *TFIDF*[88].

TFIDF atribui maior importância à frequência de um termo *(TF)* num vector e penaliza a frequência nos documentos *(IDF)*.

A função BM25 introduz novos parâmetros classificativos como o comprimento de um documento ou vector e a saturação dos termos. A introdução de novos parâmetros criou várias ramificações da função BM25. Esta árvore de variantes oferece uma larga escolha de opções, mas acarreta a dificuldade de avaliação e comparação de *performances*[89].

No estado de arte encontram-se soluções como: BM25, BM25F, BM25T, BM25-adpt, BM25L, BM25+ ou OkapiBM25 [89][87].

A fórmula de BM25 é representada pela Equação 3.6:

$$score(q,d) = \sum_{t \in q} \frac{occurs_t^d}{k1\left((1-b) + b\left(\frac{l_d}{avgl_d}\right)\right) + occurs_t^d} * IDF(t) \tag{3.6}$$

À frequência de um termo t num documento d ($occurs_q^t$), é agora adicionado um parâmetro *k1* que permite calibrar a influência desse peso no cálculo final. Além disso, a extensão de um vector ou documento (l_d) é agora um critério ponderado em relação à média de comprimento ($avgl_d$) dos documentos ou vectores de uma colecção. Também este parâmetro é calibrado pela variável *(b)*. O valor de *k1* deve respeitar uma gama de [1.2, 2.0] e $b \in [0,1]$.

A *inverse document frequency* também é alterada em relação ao seu formato aplicado em *TFIDF* como se verifica na Equação 3.7:

$$IDF(t) = log\frac{N - d_f(t) + 0.5}{df(t) + 0.5} \tag{3.7}$$

N representa o número de vectores ou documentos totais, enquanto que $d_f(t)$ contabiliza o número de documentos no qual surge o termo t.

3.4.6 BM25F

A função BM25F surge como uma extensão da BM25[90]. Enquanto que BM25 é uma função que considera a frequência de termos, frequência de documento e comprimento de documento num único campo, BM25F permite verificar múltiplos campos e considerar a estrutura de um documento ou vector.

Em primeiro lugar obtém-se o peso acumulado de um termo em todos os campos:

$$weight(t,d) = \sum_{c \in d} \frac{occurs_{t,c}^d \cdot boost_c}{((1-b_c) + b_c\frac{l_c}{avgl_c})} \tag{3.8}$$

O $boost_c$ é o factor que é aplicado ao campo c cuja extensão l_c é comparada com a média de comprimento dos vários campos $avgl_c$. Para reduzir o efeito da frequência de um termo no resultado final é aplicada uma saturação não-linear: $\frac{weight(t,d)}{k1+weight(t,d)}$.

A fórmula final resulta do produto de *IDF*, da função 3.7, com a saturação não linear com base no peso acumulado de um termo em todos os campos.

$$score(q,d) = \sum_{t \in q} IDF(t) \cdot \frac{weight(t,d)}{k1 + weight(t,d)} \tag{3.9}$$

3.4.7 BM25L

Esta variante surge com o intuito de prevenir a penalização que BM25 introduz em relação a vectores ou documentos de maior comprimento. Para tal, *Lv & Zhai*[89][91] adicionam um parâmetro regulável (δ) que tem um efeito de *shift*, na função favorecendo os vectores de menor extensão. Segundo os autores deve ser calibrado com o valor 0.5. Por forma a normalizar a influência dos termos com valor 1, multiplicam o peso de *TF* por $(k1+1)$.

$$score(q,d) = \sum_{t \in q} \frac{(k1+1) \cdot (c_{td} + \delta)}{k1 + (c_{td} + \delta))} * IDF(t) \tag{3.10}$$

onde,

$$c_{td} = \frac{occurs_t^d}{1 - b + b\left(\frac{l_d}{avdl_d}\right)} \tag{3.11}$$

e

$$IDF(t) = log\left(\frac{N+1}{d_f(t) + 0.5}\right) \tag{3.12}$$

3.4.8 BM25+

BM25+[92] é uma melhoria relativamente à versão anterior. Também com o foco em não penalizar documentos de larga extensão, BM25+ é uma função generalizada que consiste em limitar inferiormente a contribuição das ocorrências de um único termo. Contrariamente á função BM25L é adicionado δ à componente $occurs_t^d$ antes da multiplicação pelo *IDF(t)*.

Como é uma solução não só para a função BM25, *Lv & Zhai* definiram que δ deveria ter o valor 1.

$$score(q,d) = \sum_{t \in q} \left(\frac{(k1+1) \cdot occurs_t^d}{k1 \cdot \left((1-b) + b \cdot \left(\frac{l_d}{avgl_d}\right)\right) + occurs_t^d} + \delta \right) * IDF(t) \tag{3.13}$$

onde,

$$IDF(t) = log\left(\frac{N+1}{d_f(t)}\right) \tag{3.14}$$

3.4.9 BM25-Adpt

BM25-Adpt[93] é uma adaptação na qual é dada maior importância ao parâmetro de calibração *k1* na ocorrência de um termo num documento.

As versões anteriores aplicam um valor geral ao parâmetro $k1$. Nesta abordagem este parâmetro é calibrado individualmente para cada termo.

Começa-se pela probabilidade de verificar pelo menos uma ocorrência, na qual é atribuído zero, ou mais ocorrências e o query:

$$p(1|0,q) = \frac{df_r + 0.5}{N+1} \tag{3.15}$$

derivando a probabilidade de verificar mais do que uma ocorrência:

$$p(r+1|r,q) = \frac{df_{r+1} + 0.5}{df_r + 1} \tag{3.16}$$

a partir do qual o ganho de informação em qualquer ponto da função pode ser calculado como a mudança de r para r + 1 ocorrências, subtraindo a probabilidade inicial:

$$G_q^r = log_2 \frac{df_{r+1} + 0.5}{df_r + 1} - log_2 \frac{df_{tr} + 0.5}{N+1} \tag{3.17}$$

Em vez de se usar a frequência de termo ($occurs_t^d$), é definida uma nova variável baseada no resultado da normalização representada na Equação 3.11.

df_r é mais complexo e define-se por:

$$df_r = \begin{cases} |D_{t_{|}c_{td} \geq r-0.5}, & r > 1 \\ df_t, & r = 1 \\ N, & r = 0 \end{cases} \tag{3.18}$$

Em suma quando $r = 0$, o número de documentos ou vectores é totalmente considerado. Já quando $r = 1$ é a frequência de termo que é aplicada. Nos restantes casos, considera-se o número de documentos $|D_t|$, que contém o termo t com a condição de $c_{td} > r$.

Então a calibração de $k1$, neste caso, é efectuada alinhando a função de ganho de informação com a função BM25 atribuindo um termo especifico $k_1\,'$.

$$k_1\,' = arg_{k1} min \sum_{t=0}^{T} \left(\frac{G_q^r}{G_q^1} - \frac{(k_1+1)\cdot r}{k_1 + r} \right)^2 \tag{3.19}$$

Substituindo k_1 na Equação 3.6 e *IDF* pela nova variável G_q^1,

$$score(q,d) = \sum_{t \in q} G_q^1 \cdot \frac{(k_1\,' + 1)\cdot occurs_t^d}{k_1\,' \cdot ((1-b) + b \cdot (\frac{l_d}{avgl_d}) + occurs_t^d} \tag{3.20}$$

3.4.10 BM25T

É uma nova abordagem de *Lv & Zhai*[89] na qual introduzem um novo método de calcular o parâmetro k_1.

Esta metodologia inicia-se seleccionando um conjunto de documentos, C_w, que contêm um termo. Um dos focos desta abordagem passa por garantir que a contribuição da extensão da normalização da frequência de termo é proporcional à proporção dos documentos com maior contribuição da extensão da normalização da frequência de termo.

$$k_1\,' = arg_{k1} min \left(g_{k_1} - \frac{\sum_{D \in C_w} log(c_{td}) + 1}{df_t} \right)^2 \tag{3.21}$$

onde c_td é definida na Equação 3.11 e g_{k_1} é definida por:

$$g_{k_1} = \begin{cases} \frac{k_1}{k_1-1} \cdot log(k_1) & se \quad k_1 \neq 1 \\ 1 & restantes \quad casos \end{cases} \tag{3.22}$$

3.4.11 Okapi BM25

Okapi BM25 é um método, baseado no *retrieval system Okapi*, que não considera apenas a frequência dos termos presentes, mas também a extensão média de toda a colecção e o comprimento do vector ou documento em avaliação.

$$score(q,t) = \sum_{t \in q} \left[log \frac{N}{n} \right] \cdot \frac{(k+1) * occurs_t^d}{k1((1-b) + b \cdot \left(\frac{l_d}{avgl_d} \right)} \cdot \frac{(k_3+1) \cdot occurs_t^q}{(k_3 + occurs_t^q)} \tag{3.23}$$

N representa o número de documentos ou vectores presentes na colecção e n o número de documentos que contém o termo t. O parâmetro k_1 regula a influência da frequência de termo $occurs_t^d$. Se $k_1 = 0$ a frequência de termo não é contabilizada no resultado final. Se k_1 for um número elevado a frequência de termo terá uma intervenção preponderante no resultado final.

O parâmetro de calibragem do peso da extensão de um vector b, quando tem o valor $b = 0$ significa que esta variável não é considerada no resultado final. Já k_3 é o parâmetro introduzido para calibrar a influência da frequência de um termo presente no *query* $occurs_t^q$.

3.4.12 Biblioteca *rank_bm25*

Rank-BM25[94] é uma biblioteca *Python* constituída por algoritmos com funções de classificação de resultados, devolvendo de um conjunto de documentos os que mais se identificam com um *query*. É uma ferramenta útil para o desenvolvimento de *search engines*. Os algoritmos presentes são baseados na função BM25: BM25L, BM25+ e Okapi BM25.

CAPÍTULO 4

Solução Proposta

Este capítulo visa apresentar os desenvolvimentos e contribuições na elaboração de um algoritmo de processamento de texto, com base em metodologias *NLP* e técnicas classificativas para recuperação de informação num contexto de lifelog.

Na Secção 4.1 é ilustrado o modelo do algoritmo desenvolvido e a descrição do funcionamento base do mesmo. A Secção 4.2 demonstra como é elaborado o tratamento dos *queries* introduzidos no sistema. O acesso à informação armazenada na base de dados e o consequente tratamento dos mesmos é um dos temas abordados na Secção 4.3. Para além destes, a edição e adição de conceitos informativos é também uma temática considerada nesta Secção. Posteriormente, em 4.3.5, é descrito o processo de filtragem implementado no algoritmo. Já na Secção 4.4 são narradas as ferramentas *NLP* aplicadas ao tratamento de dados textuais que almejam uma expansão de termos descritivos. Por fim, a Secção 4.5 relata o processo de classificação introduzido pelo algoritmo.

4.1 Arquitectura e *workflow*

A arquitectura do algoritmo desenvolvido no âmbito desta dissertação está representada na Figura 4.1.

As ferramentas incluídas na arquitectura são direccionadas ao cumprimento das duas principais tarefas às quais o mecanismo deve responder: processamento e análise de texto e classificação de resultados.

- **Módulos: INPUT e CLEAN QUERY**

  O algoritmo começa por receber o *query* introduzido pelo utilizador, efectuando o seu processamento. O conteúdo é comunicado á base de dados, na qual está armazenada a informação proveniente do *dataset*.

  Quando os termos transportados no *query* coincidem com o conteúdo armazenado de uma imagem, são seleccionados os campos de dados desse instante e armazenados num vector.

- **Módulos: EDIT DATA e FILTERS**

  Após este processo, o sistema recorre a dados específicos dos vectores para criar, transformar e adicionar novos conteúdos de informação ao nível de localização e descrição temporal. Esta tipologia de dados é a base para o mecanismo de filtragem presente no algoritmo. Tratando-se precisamente de um crivo relativo a descrições temporais e de localização.

- **Módulo: TEXT VECTORIZATION**

  Com auxilio de metodologias *NLP*, os vectores são processados com o objectivo de ampliar o dicionário que transportam. O sistema tem incorporado um *Stemmer* e um *Lemmatizer* que são responsáveis pela geração de novos vectores. O conteúdo destes vectores é proveniente das alterações aplicadas nos termos descritivos, consequentes do processamento de ambas as ferramentas. *À posteriori*, é ainda gerado um novo vector, para cada imagem, cujo conteúdo resulta da aplicação de uma base de dados lexical. Ou seja, são armazenados termos que resultam de ramificações semânticas da informação original.

- **Módulos: RANKING e OUTPUT**

  Após o processamento de informação todos os vectores são aplicados nas funções de classificação. Neste estágio procura-se conhecer a similaridade entre a informação que o utilizador narrou e a informação que o algoritmo computacionalmente processou. O algoritmo incorpora as funções BM25 e *TFIDF* associada à similaridade de cosseno, documentadas na Secção 3.4 para a elaboração deste processo.

  Por fim, o mecanismo 4.1 devolve uma lista de resultados que resultam das iterações de classificação. As imagens com melhor *score* são as que o algoritmo classificou como as que denotam menor distância para o *query* do utilizador. Os *id* destas imagens são devolvidos pelo algoritmo no final de cada iteração.

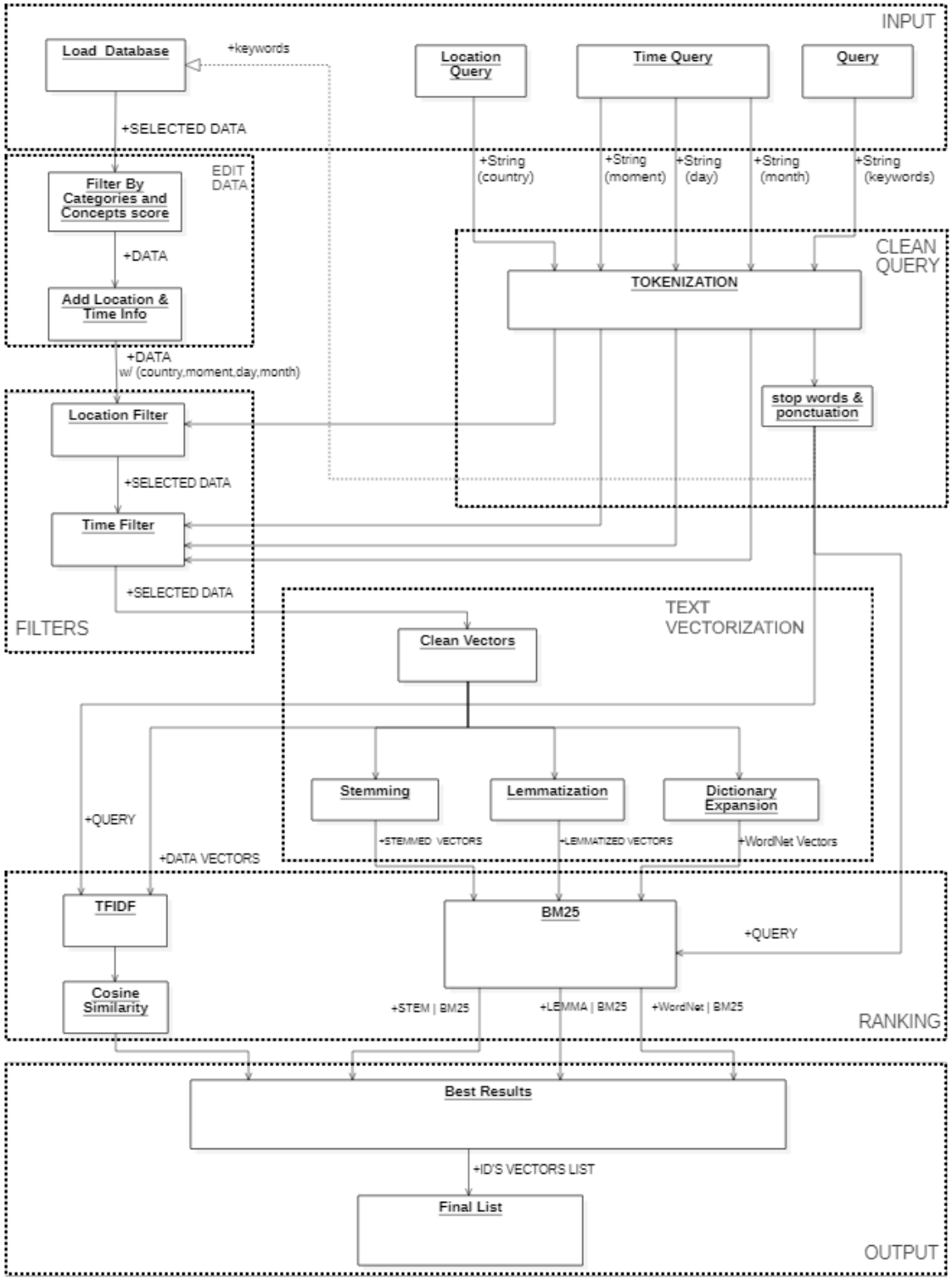


Figura 4.1: Arquitectura detalhada do algoritmo de processamento de texto e classificação de resultados.

4.1.1 HPC - Maverick2

No desenvolvimento do algoritmo, recorreu-se à tecnologia HPC para a análise de, aproximadamente, 200 mil vectores de informação, resultantes de um igual número de imagens provenientes do *dataset*. *Maverick2 (TACC)*[95] é um *cluster* HPC desenhado para suportar desenvolvimentos pesados em *Machine Learning* e *Deep Learning*. É composto por várias GPUs o que proporciona um cenário de desenvolvimento e uma *performance* avançados.

O *cluster* HPC interliga dois sistemas Lustre oferecendo ao utilizador dois directórios de trabalho. O directório $\$Home$ é indicado para actividades de compilação e edição de pequenos *scripts* de código. O segundo directório, denomina-se $\$Work$ e é indicado para o desenvolvimento de *scripts* de código extensos e permite a aplicação de longos *datasets*. A interacção com o *Maverick2* é suportada por o protocolo SSH, que permite executar acções de *login* e transferência de ficheiros. Além disso, o *Maverick2* tem associado uma documentação detalhada[1] o que permitiu rapidamente iniciar blocos de testes no algoritmo. Este *cluster* HPC oferece no seu software as bibliotecas mais comuns da linguagem de programação Python, como *NumPy, Pandas, Matplotlib, Scikit-Learn ou TensorFlow*, o que permitiu executar rapidamente os *scripts* do algoritmo. Ademais, o directório $\$Work$ permite a instalação de bibliotecas adicionais necessárias á execução dos *scripts* de código.

4.2 *Queries*

O algoritmo desenvolvido nesta dissertação é do tipo *text-based*, tipologia explorada na Secção 2.1.2. Oferece ao utilizador um ambiente para introduzir frases ou palavras chave *(keywords)* que descrevam detalhadamente um momento, para identificação e recuperação. Além disso, o *query* permite a introdução de conceitos temporais, como o momento do dia, dia da semana ou mês. Ao nível de elementos de localização o utilizador pode introduzir o país onde um momento ocorreu. Embora se possa adicionar estes conceitos como *keywords*, a introdução dos mesmos nestes campos específicos, permite que o algoritmo recorra dos mesmos para um processo de filtragem, reduzindo o leque de momentos em análise.

"Insert a Month" - Campo para o utilizador introduzir textualmente um mês.
- Exemplo: *"January"*, *"May"*, *"August"*.

"Insert a day" - Campo para o utilizador introduzir textualmente um dia.
- Exemplo: *"Monday"*, *"Saturday"*, *"Sunday"*.

"Insert location" - Campo para o utilizador introduzir textualmente um país.
- Exemplo: *"Norway"*, *"Republic of Ireland"*.

"Insert a moment" - Campo para o utilizador introduzir textualmente um momento do dia.
- Exemplo: *"morning"*, *"early morning"*, *"afternoon"*.

Na Figura 4.2 são introduzidas frases completas para a descrição do momento. No entanto, a pesquisa poderia ser efectuada apenas com base nas *keywords*: *"car", "house", "cloudy", "day", "driven", "Saturday", "August", "early afternoon"*.

[1] https://portal.tacc.utexas.edu/user-guides/maverick2

```
Search: a red car beside a white house on a cloudy day. i had driven for over an hour to get here.
 it was a Saturday in August and it was in the early afternoon.

Insert a Month: August

Insert a day: Saturday

Insert location info:

Insert a moment: early afternoon
```

Figura 4.2: Introdução de *query* no algoritmo com campos para introdução de conceitos temporais e de localização.

4.2.1 Processamento de texto

Quando introduzido no sistema, o *query* é apenas uma *string* de texto. O algoritmo necessita, por isso, de processar a informação que o mesmo transporta. Para tal, os termos que constituem o *query* são separados num processo denominado *"tokenization"*, descrito na Secção 3.1.2. Após este processo cada termo é considerado, pelo algoritmo, um elemento individual.

Se considerarmos o caso em que o utilizador apenas introduz as *keywords*, o processo é facilitado uma vez que o próprio selecciona as palavras que realmente importam para a descrição do momento. No entanto, se o *query* for semelhante ao da Figura 4.2, então é necessário uma filtragem para que o sistema seja poupado à consideração de termos que não são relevantes. Este processo é caracterizado pela eliminação de palavras recorrentes no vocabulário, de um idioma, denominadas *stopwords*. Com recurso à biblioteca *NLTK*, o algoritmo aplica a remoção de palavras que são frequentes no dicionário e gíria do idioma Inglês (Figura 4.3). Na Figura 4.4 verifica-se a remoção destes termos presentes no *query* da Figura 4.2.

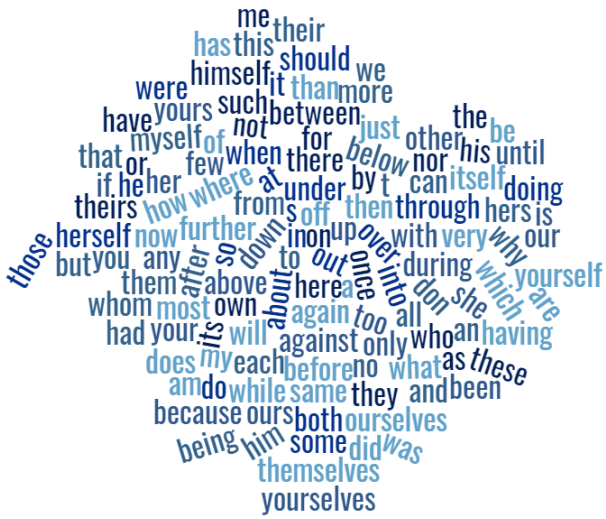


Figura 4.3: Nuvem de *stopwords* eliminadas no processamento de texto de um *query*. Tratam-se de termos frequentemente mencionados no vocabulário inglês.

```
['red', 'car', 'beside', 'white', 'house', 'cloudy', 'day', 'driven', 'hour', 'get', 'Saturday',
'August', 'early', 'afternoon']

['August']

['Saturday']

['']

['early afternoon']
```

Figura 4.4: Vectores resultantes do processamento do *query* introduzido após processamento e geração de *tokens*, bem como eliminação de *stopwords* e pontuação.
Comparando com o *query* da Figura 4.2 verifica-se que termos como *"it", "was", "an", "over"ou "a"* são eliminados .

4.3 Dataset utilizado

Os dados em análise nesta dissertação correspondem ao *dataset* do desafio *ImageCLEF*. Os dados descritivos de cada imagem são fornecidos aos participantes em formato *CSV*. São distribuídos em duas partes: *metadata* e *concepts*. A *metadata* é constituída por conceitos temporais, localização, actividades e dados biométricos. Os *concepts* são elementos que descrevem as imagens. Por uma questão de organização e facilidade de gestão da informação, nesta dissertação os dados são armazenados numa base de dados *SQL*[96].

```
minute_id,utc_time,image_path,attribute_top01,attribute_top02,attribute_top03,attribute_top04,attribute_top05,attribute_top06,
20150223_0706,UTC_2015-02-23_07:06,DATASETS/LSC2020/2015-02-23/b00000000_21i6bq_20150223_070647e.jpg,no horizon,enclosed area,
0.144,jail_cell,0.112,alcove,0.092,berth,0.074,hospital_room,0.07,bed,0.983365357,5.428367332175926 563.6330048111845 694.0870
759.649292181346,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL
20150223_0708,UTC_2015-02-23_07:08,DATASETS/LSC2020/2015-02-23/b00000001_21i6bq_20150223_070808e.jpg,no horizon,enclosed area,
light,matte,cloth,wet_bar,0.057,alcove,0.046,church/indoor,0.045,utility_room,0.042,shower,0.037,bottle,0.987764418,595.582031
0.777268946,856.0205439814815 511.2142625919058 1014.1625192901234
568.4875147795874,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NUL
20150223_0708,UTC_2015-02-23_07:08,DATASETS/LSC2020/2015-02-23/b00000002_21i6bq_20150223_070809e.jpg,enclosed area,no horizon,
lighting,glass,glossy,matte,metal,research,television_room,0.141,airplane_cabin,0.119,server_room,0.078,television_studio,0.06
186.05821585852254 648.423225308642 636.8781649376735,person,0.956759155,12.872315960165896 299.8362213323924 394.798996913580
285.51521916034795 1022.3304398148148
724.8509518962261,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NUL
20150223_0708,UTC_2015-02-23_07:08,DATASETS/LSC2020/2015-02-23/b00000003_21i6bq_20150223_070810e.jpg,enclosed area,no horizon,
lighting,glossy,glass,matte,cloth,metal,television_studio,0.165,server_room,0.095,chemistry_lab,0.075,beauty_salon,0.064,airpl
627.8402295524692 620.6660928016852,person,0.967420101,489.1369598765432 277.8320148562597 1020.7521219135803 714.867568843620
710.0898979123959,person,0.825922728,30.996413242669753 270.23500726242696 761.0757619598766
```

Figura 4.5: Retalho do ficheiro *csv* que transporta os conceitos e atributos narrativos das imagens.

(a) 20150223_070647 **(b)** 20150223_070808 **(c)** 20150223_070809 **(d)** 20150223_070810

Figura 4.6: Imagens descritas no ficheiro csv.

imageclef2020-metadata.csv

```
minute_id,utc_time,local_time,timezone,lat,lon,semantic_name,elevation,speed,heart,calories,activity_type,steps
20150223_0000,UTC_2015-02-23_00:00,2015-02-23_00:00,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0001,UTC_2015-02-23_00:01,2015-02-23_00:01,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0002,UTC_2015-02-23_00:02,2015-02-23_00:02,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0003,UTC_2015-02-23_00:03,2015-02-23_00:03,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0004,UTC_2015-02-23_00:04,2015-02-23_00:04,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0005,UTC_2015-02-23_00:05,2015-02-23_00:05,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0006,UTC_2015-02-23_00:06,2015-02-23_00:06,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0007,UTC_2015-02-23_00:07,2015-02-23_00:07,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0008,UTC_2015-02-23_00:08,2015-02-23_00:08,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0009,UTC_2015-02-23_00:09,2015-02-23_00:09,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0010,UTC_2015-02-23_00:10,2015-02-23_00:10,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0011,UTC_2015-02-23_00:11,2015-02-23_00:11,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0012,UTC_2015-02-23_00:12,2015-02-23_00:12,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0013,UTC_2015-02-23_00:13,2015-02-23_00:13,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0014,UTC_2015-02-23_00:14,2015-02-23_00:14,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0015,UTC_2015-02-23_00:15,2015-02-23_00:15,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0016,UTC_2015-02-23_00:16,2015-02-23_00:16,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0017,UTC_2015-02-23_00:17,2015-02-23_00:17,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0018,UTC_2015-02-23_00:18,2015-02-23_00:18,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0019,UTC_2015-02-23_00:19,2015-02-23_00:19,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
20150223_0020,UTC_2015-02-23_00:20,2015-02-23_00:20,Europe/Dublin,53.3892,-6.15827,Home,NULL,NULL,NULL,1.2062000036239624,NULL,NULL
```

Figura 4.7: Ficheiro *csv* com *metadata* relativa a dados biométricos, localização, actividades, entre outros.

4.3.1 Comunicação *Query* - Base de dados

Após o processamento de texto do *query* o algoritmo compara os diferentes *tokens* com o conteúdo da base de dados. Quando um campo de dados de um momento contém um termo coincidente com um termo do *query*, esse momento é seleccionado para o processamento e classificação de resultados. Os restantes momentos são descartados.

Este processo é realizado com recurso à interface de gestão de base de dados *SQlite3*[96]. Com recurso ao comando *'Select'* os campos de dados de um momento são importados e armazenados num vector.

Os atributos que detalham as imagens descrevendo os *objects* ou elementos presentes, são um dos campos nos quais é efectuada a pesquisa. Além destes, também os conceitos, categorias, tipo de actividade e localização são alvo de comparação com cada termo introduzido pelo utilizador.

4.3.2 Tipologias utilizadas

Sempre que um termo do *query* é associado a um termo presente num dos campos mencionados anteriormente, é seleccionada uma panóplia de informação que será relevante para a obtenção dos momentos em pesquisa.

Essa informação é armazenada num único vector para cada imagem. Cada vector é composto pelos seguintes elementos:

minute id - Resulta do momento em que a imagem foi obtida, mas é também o elemento de identificação da mesma.

UTC time - Data e hora de registo da imagem.

Attributes - Nos dados fornecidos no *dataset* estão presentes dez atributos que detalham cada imagem com elementos descritivos da mesma.

Category - Resultam do processamento de imagens. A cada imagem são atribuídas categorias que os algoritmos associam ás mesmas. Por exemplo, se uma imagem retratar uma cabine de avião *("airplane_cabine")*, uma das categorias poderá ser *"airplane"*. Cada categoria tem um *score* associado que indica a confiança com que a mesma foi associada

à imagem. Se o *score* tiver um valor muito baixo então é muito provável que a categoria não pertença a essa imagem.

Concepts - São elementos semelhantes aos atributos. Anotados após identificação e extracção de características de uma imagem, são também classificados com um *score*.

Timezone & **Semantic** - São os campos na base de dados com detalhes sobre localização. *Timezone* é representativo de uma cidade e o respectivo continente *(Continent/City)*. Embora o campo de dados se intitule *semantic*, na realidade, é um campo que detalha locais específicos, como uma Universidade ou um Teatro.

Activity - É o campo no qual estão armazenadas as actividades que decorrem numa imagem como: caminhar *("walking")* ou a deslocação num tipo de transporte *("transport")* ou *("airplane")*.

4.3.3 Filtragem de conceitos e categorias

A identificação ou atribuição de conceitos e categorias a uma imagem é acompanhada de um *score*. Este valor é sinónimo da certeza com que um destes campos se interliga com a imagem.

Uma vez que os dados em formato de texto serão expostos a uma classificação, para identificar possíveis similaridades com o *query*, é fundamental garantir que o algoritmo considera apenas os conceitos e categorias que verdadeiramente se relacionam com a imagem.

Considerando como exemplo a imagem 20160820_1147, ilustrada na Figura 4.9, bem como o seu vector original, representado na Figura 4.8, podem-se comparar os conceitos e categorias descritos com o que se verifica visualmente na imagem.

```
['20160820_1147', 'UTC_2016-08-20_11:47', 'no horizon', 'enclosed area', 'man-made', 'cloth'
, 'reading', 'wood', 'natural light', 'glass', 'soothing', 'plastic', 'dorm_room', '0.493', '
television_room', '0.057', 'youth_hostel', '0.037', 'hotel_room', '0.031', 'alcove', '0.021',
 'person', '0.991803646', 'tv', '0.987687886', 'book', '0.958402097', 'Rathdown Park', 'NULL'
, 'Europe/Dublin']
```

Figura 4.8: Vector inicial com a informação relativa á imagem 20160820_1147. Cada vector respeita a seguinte distribuição:
[Minute id|UTC|Attributes|Categories and categories score|Concepts and concepts score|Semantic name|Activities|Timezone].

Figura 4.9: Imagem 20160820_1147 presente no *dataset ImageCLEF*.

Tabela 4.1: Conceitos, categorias e respectivos *scores* da imagem 20160820_1147.

Categories	Score	Concepts	Score
dorm room	0.493	person	0.99
television room	0.057	tv	0.988
youth hostel	0.037	book	0.95
hotel room	0.031		
alcove	0.021		

No conjunto de categorias, da imagem acima representada, verifica-se que *"dorm room"* e *"television room"* têm um valor mais elevado. Sinaliza que há uma maior confiança na atribuição desses conceitos à imagem. Analisando visualmente a imagem, claramente se conclui que são esses os elementos que descrevem a imagem. Portanto, o algoritmo deve eliminar as restantes categorias, antes do processo de classificação. O mesmo se aplica aos conceitos da imagem.

Para tal, são definidos dois parâmetros reguláveis que geram uma filtragem de conceitos e categorias. Estes parâmetros regulam um *threshold* geral, cujo objectivo é definir um valor base, para eliminar falsos elementos.

Para definir os valores de *threshold* gerou-se várias iterações num conjunto de imagens, verificando o impacto da eliminação desses termos numa pesquisa. Com base na análise nas alterações de *score* e verificando se a imagem era seleccionada como solução definiu-se um valor mínimo capaz de garantir os termos essenciais de cada vector sem impactar negativamente o *score* e classificação da imagem.

- *Concepts & Categories Threshold*
    1. *Categories threshold:* 0.04
    2. *Concepts threshold:* 0.95

4.3.4 Processamento de dados

Nesta dissertação optou-se por aprofundar a utilização de outros tipos de dados. O foco incidiu em elementos e conceitos temporais e de localização. Como se verificou na Secção anterior, estão incluídos no vector inicial elementos temporais e de localização que podem ser explorados de forma a incluir maior detalhe no vector.

Os elementos *UTC* e *Timezone* são transformados em diferentes conceitos textuais. Através do campo *UTC* criam-se detalhes textuais como momento do dia, dia da semana e mês. Já o campo *Timezone* permite que se identifique o país no qual uma imagem foi captada. Com recurso à biblioteca *datetime*[97] e *calendar*[98] são extraídos o dia e mês por extenso da data definida no campo *UTC*.

Um elemento importante para detalhar o instante em que ocorreu um momento é a parte do dia em que o mesmo ocorreu. Para esse fim, o algoritmo recorre à informação proveniente da hora de registo da imagem para definir momentos do dia e adicioná-los aos vectores respectivos. Com base na lista de tópicos de teste e em várias interacções, o algoritmo foi calibrado com a segmentação de um dia como indica a tabela seguinte:

Tabela 4.2: Momentos de um dia considerados pelo algoritmo.

Hora UTC	Momento do dia
06:30 - 08:30	early morning
08:00 - 10:45	morning
10:45 - 12:00	morning and lunch time
12:00 - 13:30	early afternoon and lunch time
13:30 - 17:15	afternoon
17:15 - 19:00	evening
19:00 - 20:30	evening and dinner time
20:30 - 24:00	night
24:00 - 06:30	late night

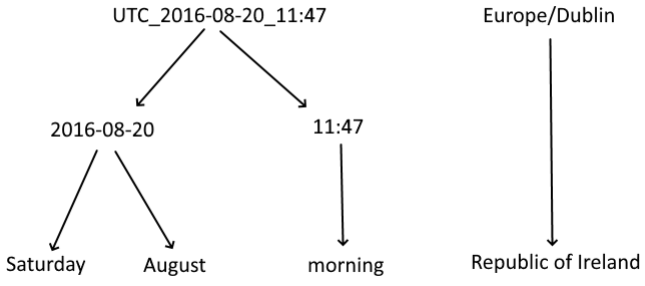


Figura 4.10: Transformação de informação UTC para conteúdo textual com recurso às bibliotecas *datetime* e *calendar*. À direita a adição de informação geográfica com recurso à biblioteca *countryinfo*[99].

```
 [['no', 'horizon', 'enclosed', 'area', 'man-made', 'cloth', 'reading', 'wood',
'natural', 'light', 'glass', 'soothing', 'plastic', 'dorm', 'room', 'television'
, 'room', 'person', 'tv', 'book', 'Rathdown', 'Park', 'Europe', 'Dublin', 'Repub
lic', 'Of', 'Ireland', 'Saturday', 'August', 'lunch', 'time', 'morning']]
```

Figura 4.11: Vector da Figura 4.8 após adição de informação e processamento de texto. Termos compostos, são agora divididos em duas palavras. Elementos como barras e *underscores* foram eliminados. Também os *scores* desaparecem após o processo de filtragem. Dados numéricos foram substituídos por informação textual.

4.3.5 Filtros

O algoritmo é constituído por um módulo de filtragem baseado em informação de tempo e localização. Por comparação directa, entre os *queries* e a informação recolhida da base de dados, este sistema permite reduzir a pesquisa a um menor leque de opções. Para esta metodologia são fundamentais os dados resultantes do processamento descrito na Secção 4.3.4. A título de exemplo introduz-se os seguintes elementos num *query*:

```
Search: airport

Insert a Month:

Insert a day:

Insert location info:

Insert a moment:
100%|                                                  | 11589/11589 [00:05<00:00, 2177.25it/s]
100%|                                                  | 11589/11589 [00:09<00:00, 1178.56it/s]
100%|                                                  | 11589/11589 [00:16<00:00, 720.88it/s]
```

Figura 4.12: *Query* sem aplicação de conceitos no módulo de filtragem.

```
Search: airport

Insert a Month: September

Insert a day: Thursday

Insert location info: Norway

Insert a moment: early morning
100%|                                                  | 160/160 [00:02<00:00, 63.60it/s]
100%|                                                  | 160/160 [00:00<00:00, 1054.60it/s]
100%|                                                  | 160/160 [00:00<00:00, 488.77it/s]
```

Figura 4.13: *Query* para processo de filtragem. Introdução dos elementos *"Norway"*,*"September"*, *"Thursday"*, *"early morning"*.

Ambas as pesquisas tencionam encontrar momentos relacionados com o termo *"airport"*. Na Figura 4.12 somente é introduzido o termo em pesquisa. Verifica-se que contrariamente ao cenário da Figura 4.13, em que são colocados elementos para o processo de filtragem, o número de vectores em análise é extremamente superior. No cenário da Figura 4.13, após o processo de filtragem são apenas considerados para análise 160 vectores. Em sentido contrário a circunstância da Figura 4.12, indica que são analisados 11589 vectores.

Além do processo de filtragem garantir uma maior probabilidade de *matching*, Figura 4.14, também a *performance* do algoritmo, num cenário de velocidade de processamento, é bastante

superior quando inseridos elementos para filtragem. Comparando os dois cenários, denota-se que com o processo de filtragem, o algoritmo realizou a sua *performance* em $2s$. Em sentido oposto, sem elementos de filtragem, o algoritmo registou $30s$ para concluir o processamento.

```
['no', 'horizon', 'enclosed', 'area', 'man-made', 'indoor', 'lighting', 'glossy'
, 'wood', 'horizontal', 'components', 'glass', 'metal', 'competing', 'airport',
'terminal', 'person', 'person', 'person', 'Oslo', '-', 'Gardermoen', 'Norway', '
International', 'Airport', 'Europe', 'Oslo', 'Norway', 'Thursday', 'September',
'early', 'morning']
['no', 'horizon', 'enclosed', 'area', 'man-made', 'indoor', 'lighting', 'glossy'
, 'working', 'glass', 'matte', 'metal', 'horizontal', 'components', 'airplane',
'cabin', 'beauty', 'salon', 'server', 'room', 'pharmacy', 'person', 'Oslo', '-',
 'Gardermoen', 'Norway', 'International', 'Airport', 'Europe', 'Oslo', 'Norway',
 'Thursday', 'September', 'early', 'morning']
['no', 'horizon', 'enclosed', 'area', 'man-made', 'indoor', 'lighting', 'wood',
'glossy', 'glass', 'horizontal', 'components', 'reading', 'matte', 'elevator', '
shaft', 'staircase', 'mezzanine', 'person', 'laptop', 'Oslo', '-', 'Gardermoen',
 'Norway', 'International', 'Airport', 'Europe', 'Oslo', 'Norway', 'Thursday', '
September', 'early', 'morning']
['no', 'horizon', 'enclosed', 'area', 'man-made', 'indoor', 'lighting', 'glossy'
, 'glass', 'wood', 'metal', 'horizontal', 'components', 'railing', 'elevator', '
shaft', 'bow', 'window', 'indoor', 'person', 'person', 'person', 'Oslo', '-', 'G
ardermoen', 'Norway', 'International', 'Airport', 'Europe', 'Oslo', 'Norway', 'T
hursday', 'September', 'early', 'morning']
['no', 'horizon', 'enclosed', 'area', 'man-made', 'natural', 'light', 'indoor',
'lighting', 'dry', 'glossy', 'natural', 'horizontal', 'components', 'scary', 'el
evator', 'shaft', 'airplane', 'cabin', 'staircase', 'car', 'interior', 'aquarium
', 'Oslo', '-', 'Gardermoen', 'Norway', 'International', 'Airport', 'Europe', 'O
slo', 'Norway', 'Thursday', 'September', 'early', 'morning']
['no', 'horizon', 'enclosed', 'area', 'man-made', 'cloth', 'indoor', 'lighting',
 'wood', 'competing', 'working', 'plastic', 'socializing', 'beauty', 'salon', 'm
artial', 'arts', 'gym', 'discotheque', 'television', 'studio', 'bowling', 'alley
', 'cup', 'person', 'cup', 'Oslo', '-', 'Gardermoen', 'Norway', 'International',
 'Airport', 'Europe', 'Oslo', 'Norway', 'Thursday', 'September', 'early', 'morni
ng']
['no', 'horizon', 'enclosed', 'area', 'man-made', 'indoor', 'lighting', 'glossy'
, 'wood', 'matte', 'cloth', 'soothing', 'vertical', 'components', 'corridor', 'b
owling', 'alley', 'Oslo', '-', 'Gardermoen', 'Norway', 'International', 'Airport
', 'Europe', 'Oslo', 'Norway', 'Thursday', 'September', 'early', 'morning']
```

Figura 4.14: Vectores após o processo de filtragem baseado em conceitos de localização e tempo. Os elementos *"Norway"*, *"September"* , *"Thursday"* e *"early morning"* estão presentes em todos os vectores que passam o processo de filtragem.

4.4 Expansão de dicionário

Recorrendo a técnicas *NLP* o vocabulário dos diversos vectores, considerados para análise, é processado com o objectivo de ampliar as ligações comunicacionais entre a informação do utilizador e a informação descritiva dos momentos armazenados. No Capítulo 3 são destacadas as metodologias aplicadas para almejar essa meta.

Cada vector, representando uma imagem, é processado em ferramentas baseadas em conceitos de *Stemming* e *Lemmatization*. Este processo garante que os termos de um vector apresentam-se de uma nova forma ampliando a probabilidade de coincidir com algum termo do *query*.

Além dos vectores originais, de *stemming* e de *Lemmatization*, é ainda gerado um novo vector com ramificações semânticas dos termos presentes no vector original. Este procedimento é baseado na ferramenta *WordNet* que gera um vector de termos com ligações semânticas aos termos originais baseado em sinónimos, hiperónimos e hipónimos.

Usufruindo da aplicação da biblioteca *NLTK* o algoritmo aplica um *Stemmer* e um *Lemmatizer* que transformam os termos em possíveis formas semelhantes às que o utilizador introduziu no *query*. As Figuras 4.15 e 4.16 mostram que a expansão de dicionário é gerada em grandes proporções, oferecendo bastantes recursos para a análise processada pelo algoritmo.

```
Original:
 [['no', 'horizon', 'enclosed', 'area', 'man-made', 'cloth', 'reading', 'wood', 'natural', 'ligh
t', 'glass', 'soothing', 'plastic', 'dorm', 'room', 'television', 'room', 'person', 'tv', 'book'
, 'Rathdown', 'Park', 'Europe', 'Dublin', 'Republic', 'Of', 'Ireland', 'Saturday', 'August', 'lu
nch', 'time', 'morning']]


Lemmatization
 [['no', 'horizon', 'enclose', 'area', 'man-made', 'cloth', 'read', 'wood', 'natural', 'light',
'glass', 'soothe', 'plastic', 'dorm', 'room', 'television', 'room', 'person', 'tv', 'book', 'Rat
hdown', 'Park', 'Europe', 'Dublin', 'Republic', 'Of', 'Ireland', 'Saturday', 'August', 'lunch',
'time', 'morning']]

Stemming:
 [['no', 'horizon', 'enclos', 'area', 'man-mad', 'cloth', 'read', 'wood', 'natur', 'light', 'gla
ss', 'sooth', 'plastic', 'dorm', 'room', 'televis', 'room', 'person', 'tv', 'book', 'rathdown',
'park', 'europ', 'dublin', 'republ', 'Of', 'ireland', 'saturday', 'august', 'lunch', 'time', 'mo
rn']]
```

Figura 4.15: Transformação e geração de novos vectores com recurso a *Porter Stemmer* e *WordNet Lemmatizer*. A redução de verbos como *"reading"* para *"read"* ou a eliminação da maiúscula inicial em *"Saturday"* para *"saturday"* são exemplos de alterações aplicadas. O algoritmo abrange ainda a possibilidade do utilizador incluir palavras incompletas como *"natur"* ao invés de *"natural"*.

```
WordNet:
 [['no', 'nobelium', 'No', 'atomic_number_102', 'no', 'no', 'no_more', 'no', 'no', 'horizon', 'apparent_horizon', 'vis
ible_horizon', 'sensible_horizon', 'skyline', 'horizon', 'view', 'purview', 'horizon', 'horizon', 'celestial_horizon',
 'envelop', 'enfold', 'enwrap', 'wrap', 'enclose', 'enclose', 'hold_in', 'confine', 'enclose', 'close_in', 'inclose',
'shut_in', 'insert', 'enclose', 'inclose', 'stick_in', 'put_in', 'introduce', 'enclosed', 'area', 'country', 'area', '
area', 'region', 'sphere', 'domain', 'area', 'orbit', 'field', 'arena', 'area', 'area', 'expanse', 'surface_area', 'ma
n-made', 'semisynthetic', 'synthetic', 'fabric', 'cloth', 'material', 'textile', 'reading', 'reading', 'reading', 'met
er_reading', 'indication', 'reading', 'reading_material', 'interpretation', 'reading', 'version', 'Reading', 'recitati
on', 'recital', 'reading', 'reading', 'meter_reading', 'read', 'read', 'say', 'read', 'read', 'scan', 'read', 'take',
'read', 'learn', 'study', 'read', 'take', 'read', 'register', 'show', 'record', 'read', 'read', 'understand', 'read',
'interpret', 'translate', 'wood', 'forest', 'wood', 'woods', 'Wood', 'Natalie_Wood', 'Wood', 'Sir_Henry_Wood', 'Sir_He
nry_Joseph_Wood', 'Wood', 'Mrs._Henry_Wood', 'Ellen_Price_Wood', 'Wood', 'Grant_Wood', 'woodwind', 'woodwind_instrumen
t', 'wood', 'wood', 'natural', 'natural', 'cancel', 'natural', 'natural', 'natural', 'natural', 'natural', 'natural',
'natural', 'instinctive', 'natural', 'raw', 'rude', 'natural', 'natural', 'born', 'innate', 'lifelike', 'natural', 'li
ght', 'visible_light', 'visible_radiation', 'light', 'light_source', 'light', 'luminosity', 'brightness', 'brightness_
level', 'luminance', 'luminousness', 'light', 'light', 'light', 'illumination', 'light', 'lightness', 'light', 'light'
, 'lighting', 'light', 'sparkle', 'twinkle', 'spark', 'light', 'light', 'Inner_Light', 'Light', 'Light_Within', 'Chris
t_Within', 'light', 'lighter', 'light', 'igniter', 'ignitor', 'light', 'illume', 'illumine', 'light_up', 'illuminate',
 'light_up', 'fire_up', 'light', 'alight', 'light', 'perch', 'ignite', 'light', 'fall', 'light', 'unhorse', 'dismount'
, 'light', 'get_off', 'get_down', 'light', 'light', 'light-colored', 'light', 'light', 'light', 'light', 'unaccented',
 'light', 'weak', 'light', 'light', 'clean', 'clear', 'light', 'unclouded', 'light', 'lightsome', 'tripping', 'light',
 'light', 'light', 'faint', 'light', 'swooning', 'light-headed', 'lightheaded', 'light', 'abstemious', 'light', 'light
', 'scant', 'short', 'light', 'light', 'idle', 'light', 'light', 'light', 'lite', 'low-cal', 'calorie-free', 'light',
'wakeful', 'easy', 'light', 'loose', 'promiscuous', 'sluttish', 'wanton', 'lightly', 'light', 'glass', 'glass', 'drink
ing_glass', 'glass', 'glassful', 'field_glass', 'glass', 'spyglass', 'methamphetamine', 'methamphetamine_hydrochloride
', 'Methedrine', 'meth', 'deoxyephedrine', 'chalk', 'chicken_feed', 'crank', 'glass', 'ice', 'shabu', 'trash', 'lookin
g_glass', 'glass', 'glass', 'glass', 'glaze', 'glass', 'glass', 'glass_in', 'glass', 'glaze', 'glass', 'glass_over', '
glaze_over', 'comfort', 'soothe', 'console', 'solace', 'soothe', 'soothing', 'assuasive', 'soothing', 'plastic', 'cred
it_card', 'charge_card', 'charge_plate', 'plastic', 'fictile', 'moldable', 'plastic', 'plastic', 'pliant', 'formative'
, 'shaping', 'plastic', 'dormitory', 'dorm', 'residence_hall', 'hall', 'student_residence', 'room', 'room', 'way', 'el
bow_room', 'room', 'room', 'board', 'room', 'television', 'telecasting', 'TV', 'video', 'television', 'television_syst
em', 'television_receiver', 'television', 'television_set', 'tv', 'tv_set', 'idiot_box', 'boob_tube', 'telly', 'goggle
_box', 'room', 'room', 'way', 'elbow_room', 'room', 'room', 'board', 'room', 'person', 'individual', 'someone', 'someb
ody', 'mortal', 'soul', 'person', 'person', 'television', 'telecasting', 'TV', 'video', 'television_receiver', 'televi
sion', 'television_set', 'tv', 'tv_set', 'idiot_box', 'boob_tube', 'telly', 'goggle_box', 'book', 'book', 'volume', 'r
ecord', 'record_book', 'book', 'script', 'book', 'playscript', 'ledger', 'leger', 'account_book', 'book_of_account', '
book', 'book', 'book', 'rule_book', 'Koran', 'Quran', "al-Qur'an", 'Book', 'Bible', 'Christian_Bible', 'Book', 'Good_B
ook', 'Holy_Scripture', 'Holy_Writ', 'Scripture', 'Word_of_God', 'Word', 'book', 'book', 'book', 'reserve', 'hold', 'b
ook', 'book', 'book', 'park', 'parkland', 'park', 'commons', 'common', 'green', 'ballpark', 'park', 'Park', 'Mungo_Par
k', 'parking_lot', 'car_park', 'park', 'parking_area', 'park', 'park', 'park', 'Europe', 'European_Union', 'EU', 'Euro
pean_Community', 'EC', 'European_Economic_Community', 'EEC', 'Common_Market', 'Europe', 'Europe', 'Dublin', 'Irish_cap
ital', 'capital_of_Ireland', 'democracy', 'republic', 'commonwealth', 'republic', 'Ireland', 'Republic_of_Ireland', 'I
rish_Republic', 'Eire', 'Ireland', 'Hibernia', 'Emerald_Isle', 'Saturday', 'Sabbatum', 'Sat', 'August', 'Aug', 'august
', 'grand', 'lordly', 'august', 'revered', 'venerable', 'lunch', 'luncheon', 'tiffin', 'dejeuner', 'lunch', 'lunch', '
time', 'clip', 'time', 'time', 'time', 'time', 'time', 'clock_time', 'time', 'fourth_dimension', 'time', 'meter', 'met
re', 'time', 'prison_term', 'sentence', 'time', 'clock', 'time', 'time', 'time', 'time', 'time', 'morning', 'morn', 'm
orning_time', 'forenoon', 'good_morning', 'morning', 'dawn', 'dawning', 'morning', 'aurora', 'first_light', 'daybreak'
, 'break_of_day', 'break_of_the_day', 'dayspring', 'sunrise', 'sunup', 'cockcrow', 'dawn', 'morning']]
```

Figura 4.16: Vector de palavras geradas após o processamento do vector da Figura 4.11 com recurso à ferramenta *WordNet*.

4.5 Classificação

Os módulos de classificação que constituem o algoritmo são reforçados por duas técnicas fundamentais: BM25 e *TFIDF*(Equação 4.1) com similaridade de cosseno. O algoritmo classifica e identifica os melhores resultados aplicando estas fórmulas matemáticas, com recurso às bibliotecas *Sklearn*[82] e *rank_bm25*, em quatro frentes.

A função *TFIDF* é aplicada ao vector original, após processamento de texto, e ao *query* do utilizador. Por fim, os vectores com os valores *TFIDF* são classificados com a aplicação do método de similaridade de cosseno (Equação 4.2). A função BM25 é um recurso que o algoritmo aplica para a comparação entre o *query* e os três vectores gerados pela expansão de dicionário.

A versão OkapiBM25 (Equação4.3) é a variante BM25 aplicada pelo algoritmo. Esta versão privilegia o número de termos de um vector, bem como a frequência com que um termo ocorre no vector e no conjunto dos vectores. Os parâmetros $k1$, b e $k3$ podem ser regulados mediante a quantidade de dados em análise.

Nesta dissertação os valores respeitam as gamas definidas pelos criadores da metodologia:

- **Parametrização OkapiBM25**

  O parâmetro $k1$ responsável por calibrar a influência da ocorrência de um termo num vector é cisado no valor 1.5.

  Uma vez que os vectores têm um elevado número de termos, o parâmetro b, responsável pela relevância da extensão de um vector tem atribuído o valor 0.75.

  Já $k3$, é o parâmetro que regula a importância dos termos de um *query*. Tem atribuído o valor 0.25.

Portanto, o algoritmo analisa quatro conjuntos de resultados. Cada classificação apresenta os melhores resultados obtidos (Figura 4.17). O resultado final é a lista global de todas as imagens que surgem nas soluções dos *rankings*. A cada imagem presente na lista é atribuído o *score* mais elevado dos diferentes métodos de classificação.

$$TFIDF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} * log \frac{|D|}{|d_j : t_i \in d_j|} \tag{4.1}$$

$$cos(\theta) = \frac{query \cdot vector_i}{\parallel query \parallel \parallel vector_i \parallel} \tag{4.2}$$

$$score(q,t) = \sum_{t \in q} \left[log \frac{N}{n} \right] \cdot \frac{(k1+1) * occurs_t^d}{k1((1-b) + b \cdot \left(\frac{l_d}{avgl_d} \right)} \cdot \frac{(k_3+1) \cdot occurs_t^q}{(k_3 + occurs_t^q)} \tag{4.3}$$

```
Search: airport corridor

Insert a Month: September

Insert a day: Thursday

Insert location info: Norway

Insert a moment: early morning
100%|| 161/161 [00:02<00:00, 62.55it/s]
100%|| 161/161 [00:00<00:00, 1078.53it/s]
100%|| 161/161 [00:00<00:00, 330.55it/s]

RANKING:

TFIDF AND COSINE:  [('20160908_0733', 0.078), ('20160908_0742', 0.08), ('20160908_0743', 0.08), ('20160908_0734
', 0.08), ('20160908_0744', 0.08), ('20160908_0739', 0.084), ('20160908_0637', 0.326), ('20160908_0758', 0.37),
 ('20160908_0753', 0.373), ('20160908_0755', 0.459)]

LEMMA AND BM25  :  [('20160908_0828', 0.009), ('20160908_0756', 0.009), ('20160908_0824', 0.009), ('20160908_08
02', 0.009), ('20160908_0817', 0.009), ('20160908_0819', 0.009), ('20160908_0637', 0.224), ('20160908_0753', 0.
227), ('20160908_0758', 0.229), ('20160908_0755', 0.239)]

STEM AND BM25   :  [('20160908_0750', 0.049), ('20160908_0742', 0.05), ('20160908_0743', 0.05), ('20160908_0724
', 0.05), ('20160908_0739', 0.05), ('20160908_0750', 0.05), ('20160908_0637', 0.224), ('20160908_0758', 0.257),
 ('20160908_0753', 0.264), ('20160908_0755', 0.267)]

WordNet AND BM25:  [('20160908_0739', 0.053), ('20160908_0707', 0.054), ('20160908_0653', 0.054), ('20160908_07
23', 0.054), ('20160908_0724', 0.055), ('20160908_0750', 0.055), ('20160908_0637', 0.221), ('20160908_0758', 0.
255), ('20160908_0755', 0.277), ('20160908_0753', 0.295)]
```

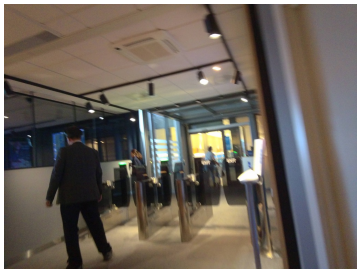
Figura 4.17: Processo de recuperação de um momento com recurso ao algoritmo desenvolvido.

O processo culmina com a apresentação da identificação das imagens e dos respectivos *scores* como ilustra a Figura 4.18. Com recurso à identificação são recuperadas as imagens presentes na Figura 4.19.

```
IMAGE MAX SCORE

1 º : ('20160908_0755', 0.459)
2 º : ('20160908_0753', 0.373)
3 º : ('20160908_0758', 0.37)
4 º : ('20160908_0637', 0.326)
5 º : ('20160908_0739', 0.084)
6 º : ('20160908_0742', 0.08)
7 º : ('20160908_0743', 0.08)
8 º : ('20160908_0734', 0.08)
9 º : ('20160908_0744', 0.08)
10 º : ('20160908_0733', 0.078)
11 º : ('20160908_0750', 0.055)
12 º : ('20160908_0724', 0.055)
13 º : ('20160908_0750', 0.055)
14 º : ('20160908_0707', 0.054)
15 º : ('20160908_0653', 0.054)
16 º : ('20160908_0723', 0.054)
17 º : ('20160908_0828', 0.009)
18 º : ('20160908_0756', 0.009)
19 º : ('20160908_0824', 0.009)
20 º : ('20160908_0802', 0.009)
21 º : ('20160908_0817', 0.009)
22 º : ('20160908_0819', 0.009)
```

Figura 4.18: Output de um processo de recuperação de momentos.

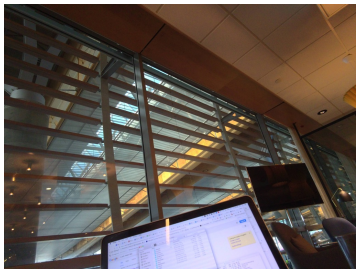


(a) TOP1

(b) TOP2

(c) TOP3

(d) TOP4



(e) TOP5

Figura 4.19: Imagens resultantes de um processo de recuperação de momentos.

CAPÍTULO 5

Resultados

Para a análise da *performance* do algoritmo, recorreu-se ao *dataset* do desafio *ImageCLEF* constituído por aproximadamente 200 mil imagens, e a uma lista de tópicos de competição do desafio *LSC 2019*. Os tópicos em análise são apresentados como o exemplo da Figura 5.1.

```
</Topic>
▼<Topic duration="180">
  <TopicID>LSC26</TopicID>
  <TopicType>expert</TopicType>
 ▼<Descriptions>
    <Description timestamp="0">A red car beside a white house.</Description>
    <Description timestamp="30">A red car beside a white house on a cloudy day.</Description>
    <Description timestamp="60">A red car beside a white house on a cloudy day. I had driven for over an hour to get here.
    </Description>
    <Description timestamp="90">A red car beside a white house on a cloudy day. I had driven for over an hour to get here. It was a
    Saturday. </Description>
    <Description timestamp="120">A red car beside a white house on a cloudy day. I had driven for over an hour to get here. It was
    a saturday in August.</Description>
    <Description timestamp="150">A red car beside a white house on a cloudy day. I had driven for over an hour to get here. It was
    a saturday in August and it was in the early afternoon.</Description>
  </Descriptions>
 ▼<RelevantImageIDs>
    <ImageID>20160820_131734_000.jpg</ImageID>
  </RelevantImageIDs>
```

Figura 5.1: Exemplo tópico de teste (tópico: LSC26).

Para teste são introduzidas, no algoritmo, as frases ou palavras chave presentes no campo $< Description >$, e activado o processo de recuperação de um momento. Como *ground truth* são utilizadas as imagens que o desafio considera como solução no campo $< RelevantImageID >$. Este é o primeiro ponto de análise do algoritmo e consiste em identificar se o mesmo detecta o momento declarado no *ground truth* (Secção 5.1).

As classificações individuais, que o algoritmo oferece, suportadas pelas técnicas BM25 e *TFIDF* com similaridade de cosseno, são posteriormente alvo de análise, verificando-se quais identificam o momento correcto e qual o *score* que lhe foi atribuído (Secção 5.5).

Por fim, os resultados obtidos são comparados com o desempenho de um sistema de recuperação de momentos,representado na Figura 5.2, denominado de *LoggyApp*[100]. Esta aplicação foi desenvolvida para o desafio *ImageCLEF*. Trata-se de um *retrieval system*, de tipologia *text-based* que oferece um contraste comparatório para a análise do algoritmo apresentado neste documento. As pesquisas por um momento no *LoggyApp* focam-se na introdução de *keywords* que conjuntamente com sistemas de filtragem permitem efectuar uma

pesquisa eficiente e rica em detalhe. A comparação com um sistema que já foi apresentado em competição, torna-se uma mais valia para retirar ilações da *performance* do algoritmo. Além disso, as abordagens de classificação diferem, o que torna este confronto, entre algoritmos, ainda mais interessante, uma vez que a classificação de resultados é um ponto crucial na performance dos *retrieval systems*.

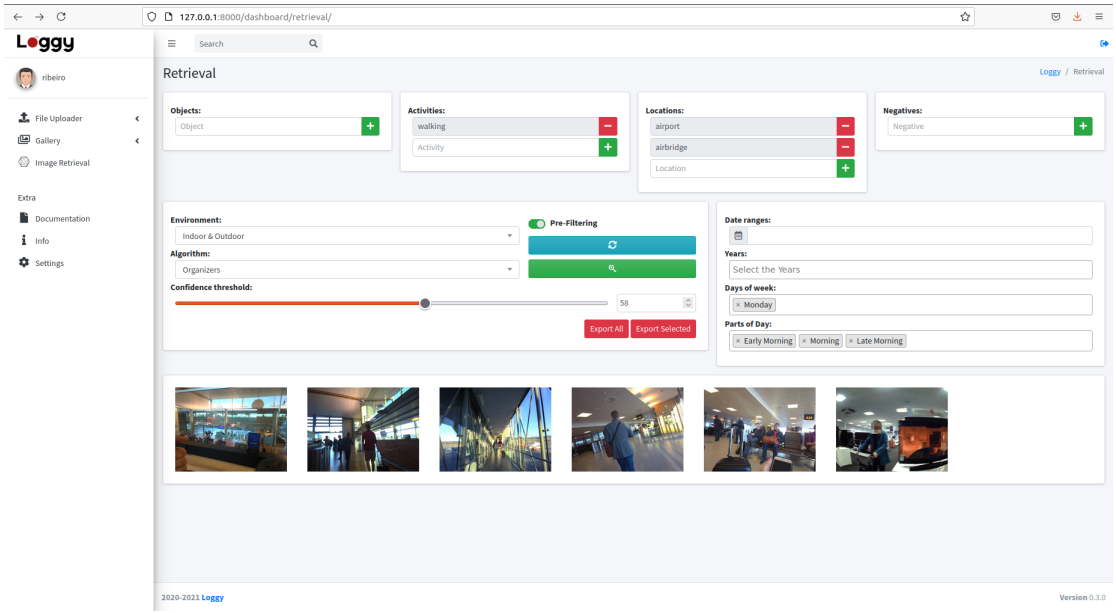


Figura 5.2: Interface do sistema LoggyApp.

5.1 Tópico LSC26

Para demonstrar o desempenho do algoritmo, num tópico do desafio LSC, destacou-se o tópico LSC26 que tem a seguinte narrativa:

- **Descrição:** *"A red car beside a white house on a cloudy day. I had driven for over an hour to get here. It was a Saturday in August and it was in the early afternoon.*
- **Keywords:** *"red, car, white, house, cloudy, day, driven"*
- **Filtros:** *August, Saturday, early afternoon*
- **RelevantImageID:** 20160820_131734

Para analisar o impacto do formato de introdução de um *query* nos resultados, aplicou-se para este tópico uma iteração com a descrição completa do mesmo, como apresentada no campo *"Descrição"* da Figura 5.1. Posteriormente, foi gerada uma nova iteração apenas com as *keywords* do tópico. Os cinco melhores resultados de cada iteração são apresentados nas Figuras 5.3 e 5.4.

Dos resultados de ambas as iterações, verifica-se de imediato que o algoritmo, num e noutro caso, recupera o momento indicado como solução. Denota-se uma ligeira discrepância de elementos em ambos os casos, uma vez que apresentam maioritariamente situações onde factualmente se verifica o elemento *"car"*, no entanto, não se verifica o elemento *"red car"*, à excepção da solução.

- ***Discrepância de elementos***

  O algoritmo é testado apenas com recurso a anotações fornecidas pelos organizadores. Ou seja, elementos como cores ou objectos específicos, habitualmente extraídos por outros algoritmos, não estão presentes nesta dissertação. Este factor pode justificar a ausência de alguns elementos introduzidos no *query*.

- ***Discrepância de classificações***

  Apesar do algoritmo filtrar alguns termos introduzidos no *query*, ao ser introduzida a descrição completa, são contabilizados, ainda assim, mais termos do que no caso da selecção de *keywords*, por parte do utilizador. Ou seja, a classificação de resultados incluirá mais termos, na situação da Figura 5.3, que podem por vezes beneficiar os resultados, ou como se verifica neste caso, prejudicar e atribuir um peso menor comparativamente á situação da Figura 5.4.

(a) TOP1: 20160820_121734 **SCORE: 0.289**

(b) TOP2: 20160820_124636 **SCORE: 0.233**

(c) TOP3: 20160820_124812 SCORE: 0.175

(d) TOP4: 20160813_124220 SCORE: 0.17

(e) TOP5: 20160820_130321 SCORE: 0.144

Figura 5.3: TOP 5 dos resultados obtidos com a introdução da descrição completa do tópico LSC26.

(a) TOP1: 20160820_131734 **SCORE: 0.326**

(b) TOP2: 20160820_124636 SCORE: 0.128

(c) TOP3: 20160820_130321 SCORE: 0.119

(d) TOP4: 20160820_125124 SCORE: 0.118

(e) TOP5: 20160820_125626 SCORE: 0.115

Figura 5.4: TOP 5 dos resultados obtidos com a introdução de *keywords* do tópico LSC26.

5.2 Tópico LSC28

- **Descrição:** *"I remember a dolls house. There were other people there, and candles too, I remember candles. There was some nice village scene in front of a lake on a picture. It was a Saturday."*
- **Keywords:** *"dolls, house, people, candles, village, lake, picture"*
- **Filtros:** *Saturday*
- **RelevantImageID:** 20160820_170806 a 20160820_171014

Na descrição deste tópico é importante realçar que é focado e detalhado um elemento descritivo, uma imagem *("picture")*, no qual, a narração indica que contém uma vila *("village")* e um lago *("lake")*. Uma das soluções ao tópico está representada na Figura 5.5.

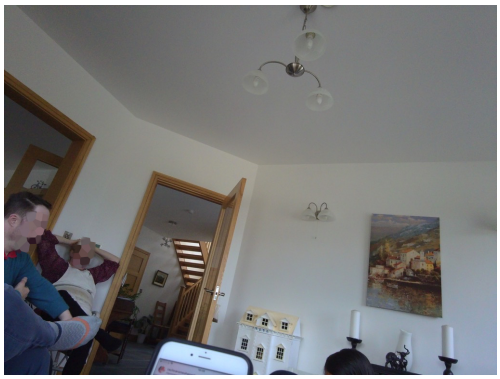


Figura 5.5: Imagem 20160820_170806, uma solução possível do tópico LSC28.

Comparando a imagem solução, deste tópico, com os resultados retornados pelo algoritmo, Figura 5.6, verifica-se desde logo, visualmente, que o momento pretendido não é recuperado.

Apesar do algoritmo não conseguir recuperar a solução, denota-se que as soluções vão ao encontro de alguns elementos descritos no *query*. Visualmente consegue-se vislumbrar, um lago *("lake")* e uma vila *("village")*. Em suma, o contexto devolvido não difere na totalidade daquilo que é descrito.

- ***Ausência de elementos descritivos***

  Um dos motivos para o algoritmo não encontrar a solução, tal como descrito no caso anterior, deve-se à ausência de elementos. Atente-se ao vector que descreve a solução presente na imagem 5.5:

Attributes: *enclosed area, no horizon, man-made, indoor lighting, wood, cloth, glass, reading, working, carpet*

Categories: *alcove,dorm room,artists loft*

Concepts: *person,person*

Os dados que constituem o vector não coincidem, maioritariamente, à narrativa do momento em pesquisa. À excepção do conceito *"person"* a descrição da imagem necessitaria da implementação de anotações, provenientes de algoritmos de processamento de imagem.

(a) TOP1: 20180512_214030 SCORE: 0.151

(b) TOP2: 20180512_214806 SCORE: 0.102

(c) TOP3: 20160827_080625 SCORE: 0.143

(d) TOP4: 20180512_214022 SCORE: 0.08

(e) TOP5: 20160820_171534 SCORE: 0.069

Figura 5.6: TOP 5 de resultados obtidos para o tópico LSC28.

5.3 Tópico LSC36

- **Descrição:** *"I was making coffee in the morning using a professional looking coffee machine. After making coffee, I was talking to someone. No, it was a few people. I remember yellow doors behind them. After a few minutes, I went back to work. It was a Monday in Norway."*
- **Keywords:** *"making, coffee, machine, talking, someone, people, yellow, doors, work"*
- **Filtros:** *"Monday, morning, Norway"*
- **RelevantImageID:** 20160905_094313 a 20160908_094657

Para a análise dos resultados obtidos para este tópico é essencial o reparo da inclusão na narrativa relativamente a descrições de momentos anteriores e posteriores. Em ambos os casos, são acrescentados elementos para além do momento que se pretende detectar.

- ***Descrição de episódios à priori e à posteriori***

  Os resultados obtidos com recurso ao algoritmo estão presentes na Figura 5.7. Verifica-se que as imagens solução não são detectadas no TOP5. O algoritmo assumiu o que o narrador relata ter efectuado após este episódio. Ou seja, as imagens que resultam desta interacção surgem claramente devido ao algoritmo equacionar o detalhe de um momento *à posteriori*. Embora as imagens correspondam correctamente ao dia, o período retornado no TOP5 é posterior ao que se pretende identificar. A informação adicional em que o narrador indica ter falado com algumas pessoas (*"I was talking to someone. No, it was a few people"*), influenciam claramente o resultado. O algoritmo só identifica uma das possíveis soluções, Figura 5.7f, no TOP20 de resultados com um *score* de 0.099.

(a) TOP1: 20160905_105347 SCORE: 0.307 **(b)** TOP2: 20160905_112158 SCORE: 0.299 **(c)** TOP3: 20160905_112406 SCORE: 0.286

(d) TOP4: 20160905_105744 SCORE: 0.277 **(e)** TOP5: 20160905_111032 SCORE: 0.274 **(f)** TOP20: 20160905_094449 **SCORE: 0.099**

Figura 5.7: Resultados obtidos para o tópico LSC36.

5.4 Tópico LSC27

- **Descrição:** *"Walking through an airbridge after a flight of about two hours in the early morning. After the airport, I immediately got a taxi to a meetting. After the airport, I immediately got a taxi to a meeting. I think it was a cloudy day on a Monday. I was in Tromso in Norway.*
- **Keywords:***"walking, airbridge, flight, airport, taxi, meeting, cloudy ,day, Tromso"*
- **Filtros:** *"early morning, Monday, Norway"*
- **RelevantImageID:** 20160905_074151 e 20160905_074223

A análise a este tópico tem o objectivo de verificar a influência do processo de filtragem na recuperação de momentos. Com base no TOP5 de resultados obtidos com o processo de filtragem (Figura 5.8), verifica-se que a imagem solução, é detectada de imediato. Sem o processo de filtragem, a solução surge com uma pior classificação. Uma vez que é considerado um maior número de imagens, surgem vectores com maior probabilidade de correspondência com o *query*. Portanto, o algoritmo atribui um menor peso á solução comparativamente à situação em que é aplicado o processo de filtragem. Em suma, o processo de filtragem incluído no algoritmo permite uma análise mais focada no exacto instante que é descrito no tópico.

Tabela 5.1: Resultados para o tópico LSC27.

TOP5(c/filtro)	Score	TOP(s/filtro)	Score
1 - 20160905_074255	0.341	1 - 20160908_072443	0.302
2 - 20160905_074151	**0.333**	2 - 20160908_073953	0.295
3 - 20160905_074327	0.162	3 - 20160908_075001	0.295
4 - 20160905_075317	0.079	4 - 20160830_082903	0.294
5 - 20160905_075349	0.067	5 - 20160830_082935	0.293
		(...)	(...)
		14 - 20160905_074151	**0.104**

(a) TOP1: 20160905_074255

(b) TOP2: 20160905_074151

(c) TOP3: 20160905_074327

(d) TOP4: 20160905_075317

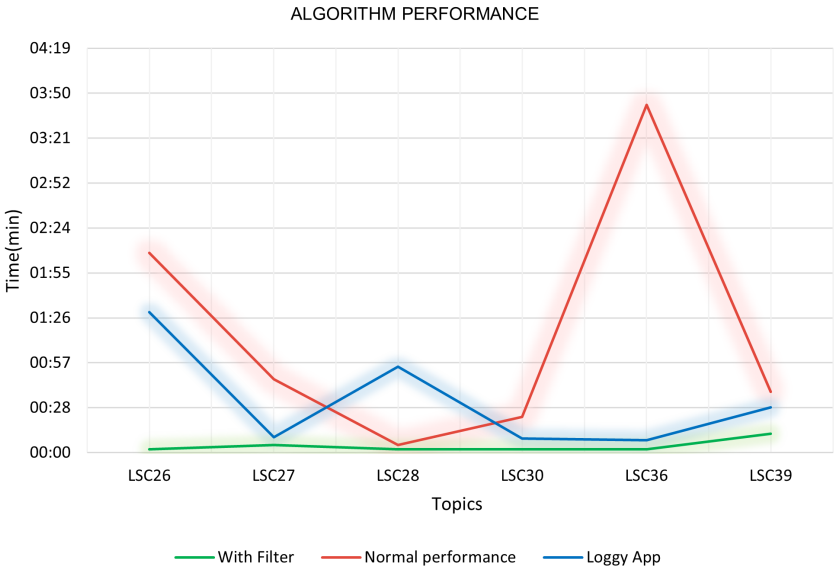
(e) TOP5: 20160905_075349

Figura 5.8: TOP 5 de resultados obtidos para o tópico LSC27 com aplicação de filtros.

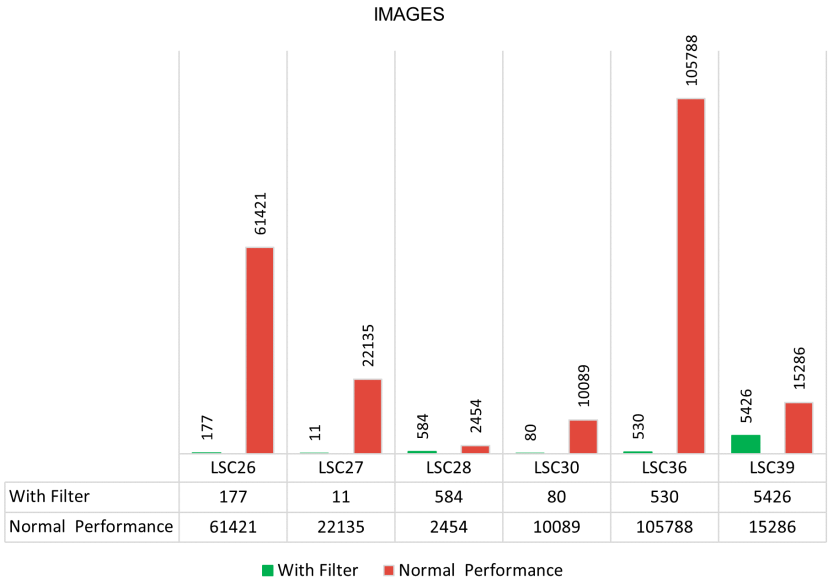
5.5 *Performance*

- ***Desempenho do algoritmo***

  Na Figura 5.9, verifica-se o desempenho do algoritmo em diferentes tópicos. Em cada iteração o algoritmo apresenta, de forma clara, uma resposta mais rápida no processamento e devolução de dados quando aplicado o processo de filtragem. A justificação prende-se na consequente redução do número de imagens. O gráfico da Figura 5.9b, demonstra precisamente, que há uma enorme discrepância entre o número de imagens numa iteração sem aplicação de crivos e uma iteração com processo de filtragem. Com o sistema de filtros, a *performance* do algoritmo regista uma gama entre 2 a 12 segundos. Estes números contrastam com os valores registados num processo de pesquisa normal, em que o máximo registado é de 3 minutos e 43 segundos.



(a) Tempo de processamento em diversos tópicos.



	LSC26	LSC27	LSC28	LSC30	LSC36	LSC39
With Filter	177	11	584	80	530	5426
Normal Performance	61421	22135	2454	10089	105788	15286

(b) Número de imagens analisadas em diferentes tópicos

Figura 5.9: Comparação da *performance* do algoritmo em diferentes tópicos de teste.

- ***Classificação***

  A Tabela 5.2 indica os resultados obtidos para as soluções de cada tópico em todas as frentes de classificação. Denota-se, numa primeira análise, que a função *Okapi* BM25 associada às metodologias de expansão de dicionário conseguem, na maior parte das situações, encontrar a solução. O facto de se renderem, uma vez que, pelo menos uma das técnicas recupera a solução, como no tópico LSC36, permite que o algoritmo adicione o momento à lista final.

  A função *TFIDF* com a classificação por similaridade de cosseno têm também um papel importante, uma vez que em algumas situações, tal como no tópico LSC30, é a

metodologia que melhor classifica a solução comparativamente às abordagens associadas à função BM25. Tal acontece em situações em que determinados termos do *query* original se encontram exactamente na mesma forma nos vectores de informação de uma imagem.

Tabela 5.2: Resultados das metodologias de classificação para diferentes tópicos teste.

		TFIDF	BM25			OUTPUT
Tópico	Solução	Cosine	Stemming	Lemmatization	WordNet	Max
LSC26	20160820_131734	0.245	-	0.326	0.17	**0.326**
LSC27	20160905_074151	0.16	0.332	0.332	0.333	**0.333**
LSC28	20160820_170806	-	-	-	-	-
LSC30	20160910_122846	0.231	0.161	0.161	0.136	**0.231**
LSC36	20160905_094413	-	0.099	-	-	**0.099**
LSC39	20160908_172759	0.266	0.102	0.102	0.086	**0.266**

Comparando as *performances* do algoritmo e do sistema *LoggyApp*, verifica-se que a aplicação tem uma velocidade de processamento menor, relativamente ao algoritmo em processo de filtragem. No entanto, o algoritmo sem qualquer crivo de informação apresenta um processamento mais lento do que o *LoggyApp*.

Quanto às classificações atribuídas, pode-se ter em consideração os tópicos LSC26 e LSC30, na Tabela 5.3, uma vez que apresentam a mesma solução. Em ambos os tópicos o *LoggyApp* apresenta um *score* mais elevado. Um factor que justifica esta diferença resulta do diferente número de imagens que permanecem após os diferentes processos de filtragem. Além disso, o *LoggyApp* pode, nesses casos, garantir ligações semânticas entre os termos disponíveis que o algoritmo é incapaz de detectar, garantindo uma confiança superior.

É ainda possível verificar que para os tópicos em teste, o algoritmo consegue detectar a solução, à excepção de um dos casos (LSC28). Como mencionado na Secção 5.2, o algoritmo não consegue solucionar a escassa descrição sobre a imagem. Já o sistema *LoggyApp* consegue encontrar o momento descrito nesse tópico. No entanto, os tópicos LSC27 e LSC36 são apenas detectados pelo algoritmo desenvolvido nesta dissertação. Uma vez que ambos os tópicos foram detectados pela técnica que engloba a expansão de dicionário associada à similaridade da função Okapi BM25, torna-se num factor preponderante comparativamente à análise por similaridade de cosseno do *LoggyApp*.

Tabela 5.3: Resultados obtidos pelo algoritmo e resultados obtidos por LoggyApp.

	Algoritmo		LoggyApp	
Tópico	Melhor Solução	Score	Melhor Solução	Score
LSC26	20160820_131734	0.326	20160820_131734	0.44
LSC27	20160905_074151	0.333	-	-
LSC28	-	-	20160820_170910	0.42
LSC30	20160910_122846	0.231	20160910_122846	0.28
LSC36	20160905_094449	0.099	-	-
LSC39	20160908_181348	0.266	20160908_180131	0.55

CAPÍTULO 6

Conclusões

No inicio desta dissertação definiu-se o objectivo de desenvolver um algoritmo capaz de interpretar uma mensagem que descreve um episódio, em linguagem natural, e recuperar esse momento. O objectivo foi cumprido na totalidade.

Comunicar com um sistema computacional e fazê-lo compreender aquilo que transmitimos é uma tarefa complexa. O sistema necessita de compreender não só os termos que habitualmente transmitimos, em linguagem natural, como é essencial que interprete as ligações que os mesmos estabelecem entre si. Esta é a chave para um sistema de recuperação de informação.

Baseado em soluções do estado de arte, de *NLP*, o algoritmo desenvolvido, foi incorporado com uma expansão de dicionário trimodal *(Stemming, Lemmatization e WordNet)*. Estas introduções aliadas a um sistema de classificação que interliga uma função matemática, ainda pouco requisitada no estado de arte *(BM25)*, com uma técnica habitual em sistemas de recuperação de momentos *(TFIDF)* e similaridade de cosseno, foram as formas definidas para reduzir o espaço comunicacional entre homem e máquina. Esta abordagem trouxe resultados satisfatórios ao nível de *performance*, classificação e recuperação de momentos.

Contudo, algumas situações não são colmatadas. Quando a narrativa aponta a um episódio anterior ou posterior, para identificar precisamente os momentos nesse intervalo, é uma questão complexa que o algoritmo não consegue almejar.

No entanto, quando o algoritmo não consegue recuperar um determinado episódio, o contexto das imagens que recupera enquadra-se com a descrição introduzida pelo utilizador. O que indica que este mecanismo compreende, ainda que com algumas ambiguidades, a mensagem do utilizador.

A introdução de um mecanismo de filtragem, baseado em conceitos de localização e tempo, impactou positivamente no desempenho do algoritmo. Não só garantiu uma *performance* mais rápida como aduziu uma eficiente recuperação de momentos.

- **Vantagens do algoritmo**

- Trata-se de um algoritmo segmentado em diferentes módulos permitindo a adição de novas metodologias. Isso engloba possíveis melhorias nas técnicas de classificação e sobretudo a introdução de novas ferramentas *NLP*.
- Permite a introdução de filtros de localização e tempo.
- A utilização de duas funções de classificação garante uma resposta frequente

- **Desvantagens do algoritmo**

- Não identifica um momento quando são detalhados os instantes anteriores e posteriores.
- Requer algum tempo de resposta em situações nas quais não são aplicados filtros.
- Não reconhece os elementos para aplicar em filtros automaticamente.
- Requer anotações mais especificas para uma melhor performance.

6.1 Trabalho Futuro

O futuro para os sistemas de recuperação de informação continuará, certamente, a dar atenção ao crescimento de dados e aos seus múltiplos formatos. Aliado a isso, o foco deve continuar na diminuição de distâncias de comunicação entre o homem e a máquina.

Portanto, o trabalho futuro que pode ser aplicado neste algoritmo é baseado nessa visão. O facto do algoritmo não reconhecer descrições de pré e pós momento, cria um fosso comunicacional. No entanto, há a possibilidade de ser resolvido com técnicas *NLP* que introduzam o reconhecimento de termos que indiquem, precisamente, um antes e um depois, como: *"After", "Before", "Afterwards"ou "Previous"*.

No mesmo sentido, o algoritmo pode ser modificado de forma a garantir o reconhecimento de *keywords*, definindo-as automaticamente como filtros. Para tal, será necessário que o mecanismo atribua a cada *keyword* uma tipologia, como localização ou conceitos temporais. Este processo, deve ser efectuado recorrendo a um modelo pré treinado, que associe os termos provenientes de um *query* a uma categoria ou conceito.

A presença de elementos não essenciais nos vectores, pode causar erros na análise de um resultado. É importante que o algoritmo faça uma correcta extracção de categorias e conceitos que não se associam a um momento, apesar de incluídos no vector. Este ponto, deve ser reforçado com uma análise ao vector e com recurso à identificação de ligações entre termos. Desta forma, estará mais próximo de identificar quais os conceitos e categorias associadas a esse momento. A metodologia de classificação pode ser alvo de diversas melhorias, que optimizem os resultados. Sobretudo, a aplicação da função BM25, deve ser revisitada na tentativa de calibrar automaticamente os seus parâmetros adequando a sua análise ao número de termos e vectores.

Por fim, a aplicação deste algoritmo num *retrieval system* seria interessante, para que o mesmo seja testado com recurso a *interfaces* gráficas e permita uma melhor experiência de teste.

Referências

[1] J. H.Turner, «Human emotions,» *Journal of Chemical Information and Modeling*, vol. 53, n.º 9, pp. 1689–1699, 2013, ISSN: 1098-6596. arXiv: arXiv:1011.1669v3.

[2] K. Diehl, G. Zauberman e A. Barasch, «How taking photos increases enjoyment of experiences,» *Journal of Personality and Social Psychology*, vol. 111, n.º 2, pp. 119–140, 2016, ISSN: 00223514. DOI: 10.1037/pspa0000055.

[3] L. Mannik, «Remembering, Forgetting, and Feeling with Photographs,» *Oral History and Photography*, pp. 77–95, 2011. DOI: 10.1057/9780230120099_5.

[4] Z. Theodosiou e A. Lanitis, «Visual Lifelogs Retrieval: State of the Art and Future Challenges,» *2019 14th International Workshop on Semantic and Social Media Adaptation and Personalization, SMAP 2019*, 2019. DOI: 10.1109/SMAP.2019.8864803.

[5] M. Aghaei, M. Dimiccoli e P. Radeva, «With whom do i interact? Detecting social interactions in egocentric photo-streams,» *Proceedings - International Conference on Pattern Recognition*, vol. 0, pp. 2959–2964, 2016, ISSN: 10514651. DOI: 10.1109/ICPR.2016.7900087. arXiv: 1605.04129.

[6] T. Yagi, K. Mangalam, R. Yonetani e Y. Sato, «Future Person Localization in First-Person Videos,» *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7593–7602, 2018, ISSN: 10636919. DOI: 10.1109/CVPR.2018.00792. arXiv: 1711.11217.

[7] A. Perina, M. Zanotto, B. Zhang e V. Murino, «Location recognition on lifelog images via a discriminative combination of generative models,» *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, p. 67, 2014. DOI: 10.5244/c.28.99.

[8] S. R. Edmunds, A. Rozga, Y. Li, E. A. Karp, L. V. Ibanez, J. M. Rehg e W. L. Stone, «Brief Report: Using a Point-of-View Camera to Measure Eye Gaze in Young Children with Autism Spectrum Disorder During Naturalistic Social Interactions: A Pilot Study,» *Journal of Autism and Developmental Disorders*, vol. 47, n.º 3, pp. 898–904, 2017, ISSN: 15733432. DOI: 10.1007/s10803-016-3002-3.

[9] L. D. Tran, M. D. Nguyen, N. T. Binh, H. Lee e C. Gurrin, «Myscéal: An Experimental Interactive Lifelog Retrieval System for LSC'20,» *LSC 2020 - Proceedings of the 3rd Annual Workshop on the Lifelog Search Challenge*, pp. 23–28, 2020. DOI: 10.1145/3379172.3391719.

[10] A. V. Mai-Nguyen, T. D. Phan, A. K. Vo, V. L. Tran, M. S. Dao e K. Zettsu, «BIDAL-HCMUS@LSC2020: An Interactive Multimodal Lifelog Retrieval with Query-to-Sample Attention-based Search Engine,» *LSC 2020 - Proceedings of the 3rd Annual Workshop on the Lifelog Search Challenge*, pp. 43–49, 2020. DOI: 10.1145/3379172.3391722.

[11] S. Heller, M. A. Parian, R. Gasser, L. Sauter e H. Schuldt, «Interactive Lifelog Retrieval with vitrivr,» em *LSC 2020 - Proceedings of the 3rd Annual Workshop on the Lifelog Search Challenge*, 2020. DOI: 10.1145/3379172.3391715.

[12] D. Khurana, A. Koli, K. Khatter e S. Singh, «Natural Language Processing: State of The Art, Current Trends and Challenges,» n.º Figure 1, 2017. arXiv: 1708.05148. URL: http://arxiv.org/abs/1708.05148.

[13] D. T. Dang-Nguyen, L. Zhou, R. Gupta, C. Gurrin e M. Riegler, «Building a disclosed lifelog dataset: Challenges, principles and processes,» 2017.

[14] C. Gurrin, A. F. Smeaton e A. R. Doherty, «LifeLogging: Personal big data,» *Foundations and Trends in Information Retrieval*, vol. 8, n.º 1, pp. 1–125, 2014, ISSN: 15540677. DOI: 10.1561/1500000033.

[15] J. Li, M. Zhang, W. Ma, Y. Liu e S. Ma, «A Multi-level Interactive Lifelog Search Engine with User Feedback,» *LSC 2020 - Proceedings of the 3rd Annual Workshop on the Lifelog Search Challenge*, pp. 29–35, 2020. DOI: 10.1145/3379172.3391720.

[16] B. Ionescu, H. Müller, R. Péteri, D. T. Dang-Nguyen, L. Piras, M. Riegler, M. T. Tran, M. Lux, C. Gurrin, Y. D. Cid, V. Liauchuk, V. Kovalev, A. Ben Abacha, S. A. Hasan, V. Datla, J. Liu, D. Demner-Fushman, O. Pelka, C. M. Friedrich, J. Chamberlain, A. Clark, A. G. S. de Herrera, N. Garcia, E. Kavallieratou, C. R. del Blanco, C. C. Rodríguez, N. Vasillopoulos e K. Karampidis, «ImageCLEF 2019: Multimedia retrieval in lifelogging, medical, nature, and security applications,» *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11438 LNCS, n.º i, pp. 301–308, 2019, ISSN: 16113349. DOI: 10.1007/978-3-030-15719-7_40.

[17] S. H. Lee, W. S. Kang e C. Moon, «Lifelog-based classification of mild cognitive impairment using artificial neural networks,» *International Conference on Electronics, Information and Communication, ICEIC 2018*, vol. 2018-January, pp. 1–2, 2018. DOI: 10.23919/ELINFOCOM.2018.8330611.

[18] A. J. Sellen e S. Whittaker, «Beyond total capture: A constructive critique of lifelogging,» *Communications of the ACM*, vol. 53, n.º 5, pp. 70–77, 2010, ISSN: 00010782. DOI: 10.1145/1735223.1735243.

[19] P. Wang, L. Sun, A. F. Smeaton, C. Gurrin e S. Yang, «Computer Vision for Lifelogging: Characterizing Everyday Activities Based on Visual Semantics,» *Computer Vision For Assistive Healthcare*, pp. 250–282, 2018. DOI: 10.1016/B978-0-12-813445-0.00009-5.

[20] M. Bolanos, M. Dimiccoli e P. Radeva, «Toward Storytelling from Visual Lifelogging: An Overview,» *IEEE Transactions on Human-Machine Systems*, vol. 47, n.º 1, pp. 77–90, 2017, ISSN: 21682291. DOI: 10.1109/THMS.2016.2616296. arXiv: 1507.06120.

[21] C. Gurrin, A. F. Smeaton e A. R. Doherty, «LifeLogging: Personal big data,» *Foundations and Trends in Information Retrieval*, vol. 8, n.º 1, pp. 1–125, 2014, ISSN: 15540677. DOI: 10.1561/1500000033.

[22] P. Soleimaninejadian, Y. Wang, H. Tong, Z. Feng, M. Zhang e Y. Liu, «THIR2 at the NTCIR-13 Lifelog-2 Task: Bridging Technology and Psychology through the Lifelog Personality, Mood and Sleep Quality,» *Proceedings of NTCIR13*, n.º 61532011, pp. 20–27, 2017. URL: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings13/pdf/ntcir/03-NTCIR13-LIFELOG-SoleimaninejadianP.pdf.

[23] L. Zhou, D. T. Dang-Nguyen e C. Gurrin, «A baseline search engine for personal life archives,» *LTA 2017 - Proceedings of the 2nd Workshop on Lifelogging Tools and Applications, co-located with MM 2017*, pp. 21–24, 2017. DOI: 10.1145/3133202.3133206.

[24] S. Mann, «Continuous lifelong capture of personal experience with eyeTap,» *CARPE'04 - Proceedings of the First ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, pp. 1–21, 2004. DOI: 10.1145/1026653.1026654.

[25] K. Aizawa, D. Tancharoen, S. Kawasaki e T. Yamasaki, «Efficient retrieval of life log based on context and content,» *CARPE'04 - Proceedings of the First ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, pp. 22–31, 2004. DOI: 10.1145/1026653.1026656.

[26] J. Gemmell, G. Bell e R. Luederby J A I M E T E E -V A N , W I L L I A M J O N E S A N D B E N J A M I N, «Rsonal Database Everything,» *Communications of the Acm*, vol. 49, n.º 1, 2006.

[27] A. R. Doherty, K. Pauly-Takacs, N. Caprani, C. Gurrin, C. J. Moulin, N. E. O'Connor e A. F. Smeaton, «Experiences of aiding autobiographical memory using the sensecam,» *Human-Computer Interaction*, vol. 27, n.º 1-2, pp. 151–174, 2012, ISSN: 07370024. DOI: 10.1080/07370024.2012.656050.

[28] C. Gurrin, Z. Qiu, M. Hughes, N. Caprani, A. R. Doherty, S. E. Hodges e A. F. Smeaton, «The smartphone as a platform for wearable cameras in health research,» *American Journal of Preventive Medicine*, vol. 44, n.º 3, pp. 308–313, 2013, ISSN: 07493797. DOI: 10.1016/j.amepre.2012.11.010.

[29] A. Leibetseder e K. Schoeffmann, «LifeXplore at the Lifelog Search Challenge 2020,» *LSC 2020 - Proceedings of the 3rd Annual Workshop on the Lifelog Search Challenge*, n.º Mmm, pp. 37–42, 2020. DOI: 10.1145/3379172.3391721.

[30] A. Duane, B. P. Jónsson e C. Gurrin, «VRLE: Lifelog Interaction Prototype in Virtual Reality: Lifelog Search Challenge at ACM ICMR 2020,» *LSC 2020 - Proceedings of the 3rd Annual Workshop on the Lifelog Search Challenge*, pp. 7–12, 2020. DOI: 10.1145/3379172.3391716.

[31] A. Alateeq, M. Roantree e C. Gurrin, «Voxento: A Prototype Voice-controlled Interactive Search Engine for Lifelogs,» *LSC 2020 - Proceedings of the 3rd Annual Workshop on the Lifelog Search Challenge*, pp. 77–81, 2020. DOI: 10.1145/3379172.3391728.

[32] O. S. Khan, M. D. Larsen, L. A. S. Poulsen, B. T. Jónsson, J. Zahálka, S. Rudinac, D. Koelma e M. Worring, «Exquisitor at the Lifelog Search Challenge 2020,» *LSC 2020 - Proceedings of the 3rd Annual Workshop on the Lifelog Search Challenge*, pp. 19–22, 2020. DOI: 10.1145/3379172.3391718.

[33] M. T. Tran, T. A. Nguyen, Q. C. Tran, M. K. Tran, K. Nguyen, V. T. Ninh, T. K. Le, H. P. Trang-Trung, H. A. Le, H. D. Nguyen, T. L. Do, V. K. Vo-Ho e C. Gurrin, «FIRST-Flexible Interactive Retrieval SysTem for Visual Lifelog Exploration at LSC 2020,» *LSC 2020 - Proceedings of the 3rd Annual Workshop on the Lifelog Search Challenge*, pp. 67–72, 2020. DOI: 10.1145/3379172.3391726.

[34] P. K. Bhagat e P. Choudhary, «Image annotation: Then and now,» *Image and Vision Computing*, vol. 80, n.º April 2019, pp. 1–23, 2018, ISSN: 02628856. DOI: 10.1016/j.imavis.2018.09.017. URL: https://doi.org/10.1016/j.imavis.2018.09.017.

[35] V. N. Murthy, S. Maji e R. Manmatha, «Automatic image annotation using deep learning representations,» *ICMR 2015 - Proceedings of the 2015 ACM International Conference on Multimedia Retrieval*, n.º May, pp. 603–606, 2015. DOI: 10.1145/2671188.2749391.

[36] J. Cao, A. Zhao e Z. Zhang, «Automatic image annotation method based on a convolutional neural network with threshold optimization,» *PLoS ONE*, vol. 15, n.º 9 September, pp. 1–21, 2020, ISSN: 19326203. DOI: 10.1371/journal.pone.0238956. URL: http://dx.doi.org/10.1371/journal.pone.0238956.

[37] G. V. API, *googlevision*. URL: https://cloud.google.com/vision.

[38] ImageCLEF, *ImageCLEF - The CLEF Cross Language Image Retrieval Track | ImageCLEF / LifeCLEF - Multimedia Retrieval in CLEF*, 2003. URL: https://www.imageclef.org/%7B%5C%7D0Ahttp://www.imageclef.org/.

[39] NTCIR, *NTCIR16-Lifelog*. URL: http://ntcir-lifelog.computing.dcu.ie/.

[40] VBS, *Video Browser Showdown – The Video Retrieval Competition – Video Browser Showdown*. URL: https://videobrowsershowdown.org/.

[41] LSC, *HOME _ LSC'21 at ICMR'21*. URL: http://lsc.dcu.ie/.

[42] G. Kovalčík, V. Škrhak, T. Souček e J. Lokoč, «VIRET Tool with Advanced Visual Browsing and Feedback,» *LSC 2020 - Proceedings of the 3rd Annual Workshop on the Lifelog Search Challenge*, pp. 63–66, 2020. DOI: 10.1145/3379172.3391725.

[43] P. H. Kim e F. Giunchiglia, «Lifelog Data Model and Management : Study on Research Challenges LIFELOG DATA MODEL AND MANAGEMENT : STUDY ON RESEARCH CHALLENGES Pil Ho Kim , Fausto Giunchiglia June 2012 Technical Report # DISI-12-019 International Journal of Computer Information System,» n.º December 2013, 2015.

[44] T. D. Truong, V. T. Nguyen, T. Dinh-Duy e M. T. Tran, «Lifelogging retrieval based on semantic concepts fusion,» *LSC 2018 - Proceedings of the 2018 ACM Workshop on the Lifelog Search Challenge, co-located with ICMR 2018*, pp. 24–29, 2018. DOI: 10.1145/3210539.3210545.

[45] L. Rossetto, M. Baumgartner, N. Ashena, F. Ruosch, R. Pernischová e A. Bernstein, «LifeGraph: A Knowledge Graph for Lifelogs,» *LSC 2020 - Proceedings of the 3rd Annual Workshop on the Lifelog Search Challenge*, pp. 13–17, 2020. DOI: 10.1145/3379172.3391717.

[46] R. Schank e A. Kass, «Natural Language Processing:What's Really Involved,» 1987.

[47] Y. Tsvetkov, V. Prabhakaran e R. Voigt, «Socially responsible natural language processing,» *The Web Conference 2019 - Companion of the World Wide Web Conference, WWW 2019*, p. 1326, 2019. DOI: 10.1145/3308558.3320097.

[48] N. Madnani, «Getting started on natural language processing with Python,» *XRDS: Crossroads, The ACM Magazine for Students*, vol. 13, n.º 4, p. 5, 2007, ISSN: 1528-4972. DOI: 10.1145/1315325.1315330.

[49] V. Arnaoudova, S. Haiduc, A. Marcus e G. Antoniol, «The Use of Text Retrieval and Natural Language Processing in Software Engineering,» *Proceedings - International Conference on Software Engineering*, vol. 2, pp. 949–950, 2015, ISSN: 02705257. DOI: 10.1109/ICSE.2015.301.

[50] P. Wiriyathammabhum, D. Summers-Stay, C. Fermüller e Y. Aloimonos, «Computer Vision and Natural Language Processing,» *ACM Computing Surveys*, vol. 49, n.º 4, pp. 1–44, 2017, ISSN: 0360-0300. DOI: 10.1145/3009906.

[51] P. Silva, C. Gonçalves, C. Godinho, N. Antunes e M. Curado, «Using natural language processing to detect privacy violations in online contracts,» *Proceedings of the ACM Symposium on Applied Computing*, pp. 1305–1307, 2020. DOI: 10.1145/3341105.3375774.

[52] T. Woodward, «Information Retrieval Using Micros,» *Aslib Proceedings*, vol. 41, n.º 4, pp. 157–162, 1989, ISSN: 0001253X. DOI: 10.1108/eb051135.

[53] P. W. Cheng, S. Chennuru, S. Buthpitiya e Y. Zhang, «A language-based approach to indexing heterogeneous multimedia lifelog,» *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction, ICMI-MLMI 2010*, pp. 8–10, 2010. DOI: 10.1145/1891903.1891937.

[54] T. Publishers e P. Papers, «It is extremely difficult to define how we would ever know that a system actually "understands" language.,» *Database*, pp. 1218–1222,

[55] M. M.Ceccato and Kiyavitskaya, Zeni e Berry, «Ambiguity Identification and Measurement in Natural Language Texts,» *Technical Report DIT-04-111*, n.º December, 2004.

[56] A. B. P. R. Ray V. Harish e S. Sarkar, «Part of Speech Tagging and Local Word Grouping Techniques for Natural Language Parsing in Hindi,» *In Proceedings of ICON 2003*, 2003.

[57] A. Ritter, C. Sam, Mausam e O. Etzioni, «Low-Resource Domains Twitter,» *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1524–1534, 2011.

[58] J. Yi, T. Nasukawa, R. Bunescu e W. Niblack, «Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques,» *Proceedings - IEEE International Conference on Data Mining, ICDM*, n.º March 2015, pp. 427–434, 2003, ISSN: 15504786. DOI: 10.1109/icdm.2003.1250949.

[59] S. Sharma, P. Srinivas e R. C. Balabantaray, «Emotion Detection using Online Machine Learning Method and TLBO on Mixed Script,» *In Proceedings of Language Resources and Evaluation Conference*, p. 47.53, 2013. URL: https://interop2016.github.io/pdf/INTEROP-11.pdf.

[60] S. R. Sirsat, V. Chavan e H. S. Mahalle, «Strength and Accuracy Analysis of Affix Removal Stemming Algorithms,» *International Journal of Computer Science and Information Technologies*, vol. 4, n.º 2, pp. 265–269, 2013.

[61] V. Balakrishnan e L.-Y. Ethel, «Stemming and Lemmatization: A Comparison of Retrieval Performances,» *Lecture Notes on Software Engineering*, vol. 2, n.º 3, pp. 262–267, 2014, ISSN: 23013559. DOI: 10.7763/lnse.2014.v2.134.

[62] P. Willett, «The Porter stemming algorithm: Then and now,» *Program*, vol. 40, n.º 3, pp. 219–223, 2006, ISSN: 00330337. DOI: 10.1108/00330330610681295.

[63] H. Liu, T. Christiansen, W. A. Baumgartner e K. Verspoor, «BioLemmatizer: A lemmatization tool for morphological processing of biomedical text,» *Journal of Biomedical Semantics*, vol. 3, n.º 1, p. 3, 2012, ISSN: 20411480. DOI: 10.1186/2041-1480-3-3. URL: http://www.jbiomedsem.com/content/3/1/3.

[64] T. Mikolov, I. Sutskever, K. Chen, G. Corrado e J. Dean, «Distributed representations ofwords and phrases and their compositionality,» *Advances in Neural Information Processing Systems*, pp. 1–9, 2013, ISSN: 10495258. arXiv: 1310.4546.

[65] V. Zolotov e D. Kung, «Analysis and Optimization of fastText Linear Text Classifier,» n.º February, 2017. arXiv: 1702.05531. URL: http://arxiv.org/abs/1702.05531.

[66] A. Znotiņš, «Word embeddings for Latvian natural language processing tools,» *Frontiers in Artificial Intelligence and Applications*, vol. 289, pp. 167–173, 2016, ISSN: 09226389. DOI: 10.3233/978-1-61499-701-6-167.

[67] Y. Zhang, R. Jin e Z. H. Zhou, «Understanding bag-of-words model: A statistical framework,» *International Journal of Machine Learning and Cybernetics*, vol. 1, n.º 1-4, pp. 43–52, 2010, ISSN: 18688071. DOI: 10.1007/s13042-010-0001-0.

[68] George A. Miller, *WordNet: A Lexical Database for English*, 1995.

[69] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross e K. J. Miller, «Introduction to wordnet: An on-line lexical database,» *International Journal of Lexicography*, vol. 3, n.º 4, pp. 235–244, 1990, ISSN: 09503846. DOI: 10.1093/ijl/3.4.235.

[70] W. Senses, «Word Senses and WordNet,» 2020.

[71] P. Software Foundation, *The Python tutorial*, 2019. URL: https://docs.python.org/3/tutorial/%7B%5C%%7D0Ahttps://docs.python.org/3/tutorial/index.html.

[72] A. Drozd e A. Gladkova, «Python , Performance , and Natural Language Processing,»

[73] S. Bird, *Natural language processing with python*, O'REILLY, ed. 2009, ISBN: 9780596516499.

[74] S. Bird e E. Loper, «: The Natural Language Toolkit,» *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions* -, 31–es, 2004. URL: http://portal.acm.org/citation.cfm?doid=1219044.1219075.

[75] K. Goyal, *Top 7 Python NLP Libraries [And Their Applications in 2021]*, 2021. URL: https://www.upgrad.com/blog/python-nlp-libraries-and-applications/.

[76] R. Řehůřek e P. Sojka, «Gensim — Statistical Semantics in Python,» vol. 6611, n.º May 2010, p. 6611, 2011.

[77] Spacy, *spaCy 101_ Everything you need to know · spaCy Usage Documentation*. URL: https://spacy.io/usage/spacy-101.

[78] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard e D. McClosky, «The Stanford CoreNLP Natural Language Processing Toolkit,» n.º January 2014, pp. 55–60, 2015. DOI: 10.3115/v1/p14-5010.

[79] C. Kaur e A. Sharma, «Twitter sentiment analysis on Coronavirus using Textblob,» *EasyChair preprint*, vol. 2974, n.º March, pp. 1–10, 2020.

[80] Pynlpl, *Welcome to PyNLPl's documentation! — PyNLPl 1*. URL: https://pynlpl.readthedocs.io/en/latest/.

[81] T. De Smedt e W. Daelemans, «Pattern for python,» *Journal of Machine Learning Research*, vol. 13, n.º June 2012, pp. 2063–2067, 2012, ISSN: 15324435.

[82] Sklearn, *sklearn*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.

[83] B. Das e S. Chakraborty, «An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation,» 2018. arXiv: 1806.06407. URL: http://arxiv.org/abs/1806.06407.

[84] P. D. Arnesia e S. Madenda, «Matching images with textual document using TFIDF method,» *2012 5th International Congress on Image and Signal Processing, CISP 2012*, n.º October, pp. 1283–1289, 2012. DOI: 10.1109/CISP.2012.6469720.

[85] J. Ramos, «Using TF-IDF to Determine Word Relevance in Document Queries,» *Proceedings of the first instructional conference on machine learning*, vol. 242, n.º 1, pp. 29–48, 2003.

[86] S. Qaiser e R. Ali, «Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,» *International Journal of Computer Applications*, vol. 181, n.º 1, pp. 25–29, 2018. DOI: 10.5120/ijca2018917395.

[87] Y. Champclaux, T. Dkaki e J. Mothe, «Enhancing high precision by combining Okapi BM25 with structural similarity in an information retrieval system,» *ICEIS 2009 - 11th International Conference*

on Enterprise Information Systems, Proceedings, vol. ISAS, n.º February 2015, pp. 279–285, 2009. DOI: 10.5220/0002017202790285.

[88] K. technology, *Understanding TF-IDF and BM25 – KMW Technology*. URL: http://www.kmwllc.com/index.php/2020/03/20/understanding-tf-idf-and-bm25/.

[89] A. Trotman, A. Puurula e B. Burgess, «Improvements to BM25 and language models examined,» *ACM International Conference Proceeding Series*, vol. 27-28-Nove, pp. 58–65, 2014. DOI: 10.1145/2682862.2682863.

[90] K. M. Svore e C. J. Burges, «A machine learning approach for improved BM25 retrieval,» *International Conference on Information and Knowledge Management, Proceedings*, n.º June, pp. 1811–1814, 2009. DOI: 10.1145/1645953.1646237.

[91] Y. Lv e C. X. Zhai, «When documents are very long, BM25 fails,» *SIGIR'11 - Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, n.º I, pp. 1103–1104, 2011. DOI: 10.1145/2009916.2010070.

[92] Y. Lv e C. Zhai, «Lower-bounding term frequency normalization,» *International Conference on Information and Knowledge Management, Proceedings*, pp. 7–16, 2011. DOI: 10.1145/2063576.2063584.

[93] Y. Lv e C. X. Zhai, «Adaptive term frequency normalization for BM25,» *International Conference on Information and Knowledge Management, Proceedings*, pp. 1985–1988, 2011. DOI: 10.1145/2063576.2063871.

[94] Rank_bm25, *rank-bm25 · PyPI*. URL: https://pypi.org/project/rank-bm25/.

[95] Tacc, *TACC Stampede User Guide - TACC User Portal*. URL: https://portal.tacc.utexas.edu/user-guides/stampede.

[96] Python.org, *sqlite3 — DB-API 2*. URL: https://docs.python.org/3/library/sqlite3.html.

[97] Python Software Foundation, *datetime — Basic date and time types*, 2020. URL: https://docs.python.org/3/library/datetime.html.

[98] Calendar, *calendar — General calendar-related functions — Python 3*. URL: https://docs.python.org/3/library/calendar.html.

[99] Countryinfo, *countryinfo · PyPI*. URL: https://pypi.org/project/countryinfo/.

[100] R. Ribeiro, «UA. PT Bioinformatics at ImageCLEF 2020: Lifelog Moment Retrieval Web based Tool.,» *CLEF (Working Notes)*, 2020.

[101] T. Advanced, «Intro to HPC @ TACC Documentation,» 2020.

Appendix A

Este *Appendix* oferece uma visão geral do sistema de *high performance computer* fornecido para o desenvolvimento do algoritmo retratado nesta dissertação. Neste tópico demonstra-se uma arquitectura base de um sistema HPC e enfatiza-se os meios de comunicação com recurso ao protocolo SSH apara acesso e interacção com o HPC Maverick2.

HPC - High Computer Performance

Como ilustra a Figura 1 a arquitectura geral de um sistema HPC inclui *"Compute nodes"* e *"Login nodes"*. Estes últimos são essenciais para gerir os múltiplos utilizadores do sistema, portanto não é recomendável que se recorra a este espaço para executar aplicações. Ao invés disso, deve-se recorrer aos nós computacionais, equipados com amplas memórias e GPU's de alta *performance*.

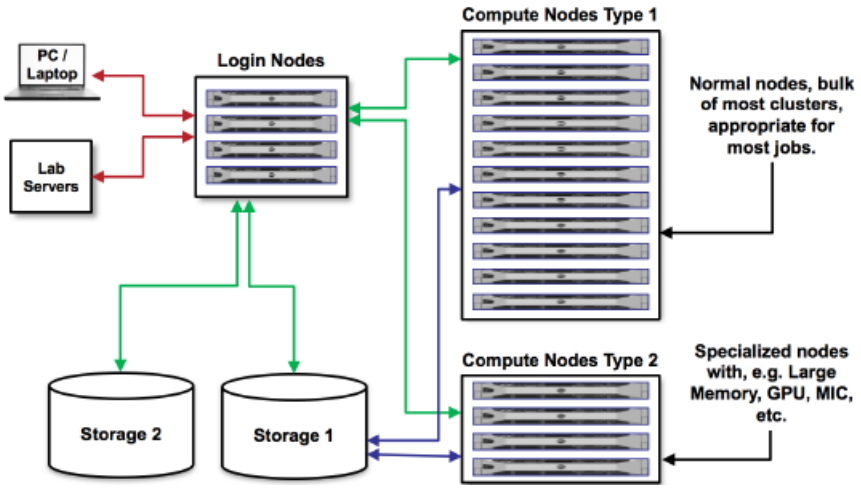


Figura 1: Arquitectura base de um sistema HPC [101].

Maverick 2

O HPC TACC Maverick2[95], contém as características presentes na Figura 2.

Model:	Super Micro X10DRG-Q Motherboard
Processor:	Intel(R) Xeon(R) CPU E5-2620 v4
Total processors per node:	2
Total cores per processor:	8
Total cores per node:	16
Hardware threads per core:	2
Hardware threads per node:	32
Clock rate:	2.10GHz
RAM:	128 GB
L1/L2/L3 Cache:	512KiB / 2MiB / 20 MiB
Local storage:	150.0 GB (~60 GB free)
GPUs:	4 x NVidia 1080-TI GPUs

(a) Maverick2 Computer nodes GTX 100

Model:	Dell PowerEdge R740
Processor:	Xeon(R) Platinum 8160 CPU @ 2.10GHz
Total processors per node:	2
Total cores per processor:	24
Total cores per node:	48
Hardware threads per core:	2
Hardware threads per node:	96
Clock rate:	2.10GHz
RAM:	192 GB
L1/L2/L3 Cache:	1536KiB / 24576KiB / 33792KiB
Local storage:	119.5 GB (~32 GB free)
GPUs:	2 NVidia V100 adapters

(b) Maverick2 Computer nodes V100

Model:	Dell PowerEdge R740
Processor:	Xeon(R) Platinum 8160 CPU @ 2.10GHz
Total processors per node:	2
Total cores per processor:	24
Total cores per node:	48
Hardware threads per core:	2
Hardware threads per node:	96
Clock rate:	2.10GHz
RAM:	192 GB
L1/L2/L3 Cache:	1536KiB / 24576KiB / 33792KiB
Local storage:	119.5 GB (~32 GB free)
GPUs:	2 NVidia P100 adapters

(c) Maverick2 Computer nodes P100

Figura 2: Caracteristicas dos *Computer Nodes* do HPC Maverick2[101]

O sistema de ficheiros apresentado no HPC Maverick2 é baseado na segmentação apresentada na Figura 3. O HPC interliga dois sistemas de ficheiros Lustre, em que cada utilizador tem dois directórios essenciais a si associados: $\$HOME$ e $\$WORK$.

- $HOME$
  - Sistema de ficheiros Lustre.
  - Capacidade geral de 1PB
  - Cada utilizador tem a si associados 10 GB.

Este directório é recomendado para actividades de compilação e edição de pequenos *scripts* de código.

- $WORK$
  - Sistema de ficheiros Lustre.
  - Capacidade geral de 20 PB
  - Cada utilizador pode utilizar 1 TB.

No directório $WORK$ é possivel a instalação de *softwares* e a execução de longos *scripts* de código. É também o directório indicado para a aplicação de extensos *datasets*.

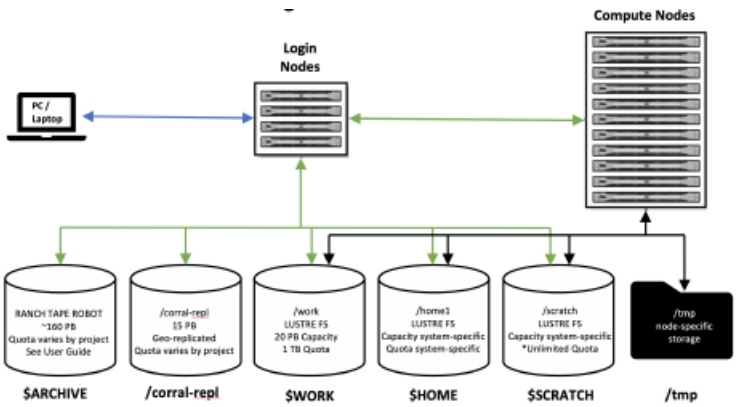


Figura 3: Cluster HPC TACC[101].

Interação

O ambiente $STOCKYARD$, exemplificado na Figura 4 é o estágio superior de toda a hierarquia. É a partir deste que se consegue aceder aos HPC TACC e aos respectivos directórios, neste caso, Maverick2(*work e home*).

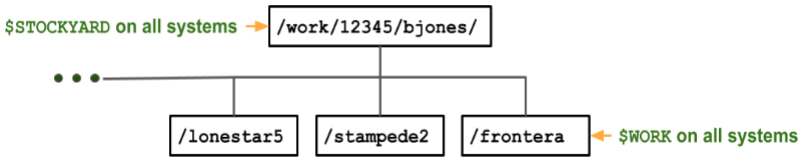


Figura 4: $STOCKYARD$ no sistema HPC TACC [95].

O protocolo de comunicação *SSH* serve de suporte à interface de comunicação com o sistema Maverick2. Este protocolo além de permitir o acesso ao sistema, inclui ferramentas para a transferência de ficheiros.

Tabela 1: Código ssh para login no HPC Maverick2.

localhost$ **ssh *username*@maverick2.tacc.utexas.edu**

- ***Login***

```
Open the application 'Terminal'
ssh username@stampede2.tacc.utexas.edu
(enter password)
(enter 6-digit token)
```

Figura 5: Ambiente de *login* para HPC TACC.

O utilizador é registado com um *username* que posteriormente é associado a um projecto alojado numa das máquinas. O *login* é efectuado com recurso ao *username* que providencia o registo no PORTAL TACC para a exposição de duvidas e problemas.

- ***Transferência de ficheiros com recurso a scp e rsync***

Tabela 2: Código scp para transferência de ficheiros.

localhost$ **scp ./myfile username@maverick2.tacc.utexas.edu: /work/01234/username/maverick2**

Tabela 3: Código rsync para transferência de ficheiros.

localhost$ **scp mybigfile username@maverick2.tacc.utexas.edu: /work/01234/username/maverick2**

Contrariamente ao *scp*, *rsync* permite sincronizar parcelas de ficheiros modificados, garantindo que não é necessário transferir todo o conteúdo novamente.

- ***Acesso a diretórios*** $\$WORK$ ***e*** $\$HOME$

Tabela 4: Acesso a directório $\$HOME$.

cd ou cdh - **cd** $\$HOME$

Contrariamente ao *scp*, *rsync* permite sincronizar parcelas de ficheiros modificados, garantindo que não é necessário transferir todo o conteúdo novamente.

Tabela 5: Acesso a directório $\$WORK$.

cd ou cdw - **cd** $\$WORK$

- ***Outros comandos***

Command	Effect
`pwd`	print working directory
`ls`	list files and directories
`ls -l`	list files in column format
`mkdir dir_name/`	make a new directory
`cd dir_name/`	navigate into a directory
`rmdir dir_name/`	remove an empty directory
`rm -r dir_name/`	remove a directory and its contents
`tree`	list files and directories hierarchically
`.` or `./`	refers to the present location
`..` or `../`	refers to the parent directory

(a) Comandos para manipulação de directórios e ficheiros.

Command	Effect
`touch file_name`	create a new file
`rm file_name`	remove a file
`rm -r dir_name/`	remove a directory and its contents
`mv file_name dir_name/`	move a file into a directory
`mv old_file new_file`	change the name of a file
`mv old_dir/ new_dir/`	change the name of a directory
`cp old_file new_file`	copy a file
`cp -r old_dir/ new_dir/`	copy a directory
`<Tab>`	autocomplete file or folder names
`<UpArrow>`	cycle through command history

(b) Comandos de exploração de conteúdo dos ficheiros.

Figura 6: Comandos interactivos para HPC TACC[101].

Software

Maverick2 oferece um vasto leque de bibliotecas para testar as aplicações desenvolvidas. No entanto, os pacotes de ferramentas são limitados para reduzir o espaço ocupado. O *cluster* HPC fornece um conjunto de bibliotecas Python frequentemente aplicadas como *NumPy, Pandas, Matplotlib, Scikit-Learn ou TensorFlow.* Para colmatar a ausência de uma biblioteca, o utilizador pode, no directório $\$Work$, instalar qualquer biblioteca adicionais essenciais à execução dos seus *scripts* de código.

```
login1$ module avail

------------------- /opt/apps/intel18/impi18_0/modulefiles -------------------
   boost/1.66                  phdf5/1.10.4    (D)
   fftw3/3.3.6                 pnetcdf/1.8.1
   parallel-netcdf/4.3.3.1     python2/2.7.16 (L,D)
   parallel-netcdf/4.6.2  (D)  python3/3.7.0  (D)

---------------------- /opt/apps/intel18/modulefiles ----------------------
   hdf5/1.8.16          mkl-dnn/0.18.1  netcdf/4.3.3.1      python3/3.7.0
   hdf5/1.10.4 (D)      nco/4.6.9       netcdf/4.6.2   (D)  udunits/2.2.25
   impi/18.0.2 (L)      ncview/2.1.7    python2/2.7.16

------------------------- /opt/apps/modulefiles -------------------------
   TACC          (L)    gcc/7.1.0                matlab/2019a          (D)
   autotools/1.2 (L)    gcc/7.3.0         (D)    mcr/9.5
   cmake/3.8.2          git/2.24.1        (L)    mcr/9.6               (D)
   cmake/3.10.2         hwloc/1.11.2             ncl_ncarg/6.3.0
   cmake/3.16.1  (L,D)  idev/1.5.5               settarg
   cuda/8.0      (g)    intel/16.0.3             swr/18.3.3
   cuda/9.0      (g)    intel/17.0.4             tacc-singularity/2.6.0
   cuda/9.2      (g,D)  intel/18.0.2       (L,D)  tacc-singularity/3.4.2 (D)
   cuda/10.0     (g)    launcher_gpu/1.0         tacc_tips/0.5
   cuda/10.1     (g)    lmod                     xalt/2.6.12           (L)
   gcc/5.4.0            mathematica/12.0
   gcc/6.3.0            matlab/2018b

  Where:
   D:  Default Module
   L:  Module is loaded
   g:  built for GPU
```

Figura 7: *Software* presente em Maverick2 [101].

Appendix B

```
IMAGE MAX SCORE

1 º : ('20160820_131734_000.jpg', 0.326)
2 º : ('20160820_124636_000.jpg', 0.128)
3 º : ('20160820_130321_000.jpg', 0.119)
4 º : ('20160820_125124_000.jpg', 0.118)
5 º : ('20160820_125625_000.jpg', 0.115)
6 º : ('20160820_124812_000.jpg', 0.11)
7 º : ('20160820_124220_000.jpg', 0.109)
8 º : ('20160813_131130_000.jpg', 0.108)
9 º : ('20160820_124708_000.jpg', 0.106)
10 º : ('20160820_124740_000.jpg', 0.105)
11 º : ('20160820_131700_000.jpg', 0.091)
12 º : ('20160827_120446_000.jpg', 0.088)
13 º : ('20160813_122141_000.jpg', 0.087)
14 º : ('20160827_132846_000.jpg', 0.084)
15 º : ('20160827_124129_000.jpg', 0.079)
16 º : ('20160820_122951_000.jpg', 0.078)
17 º : ('20160827_130545_000.jpg', 0.077)
18 º : ('20160813_132810_000.jpg', 0.064)
19 º : ('20160813_123734_000.jpg', 0.064)
```

Figura 8: Resultados do tópico LSC26 obtidos pelo algoritmo.

```
IMAGE MAX SCORE

1 º : ('20160905_074255_000.jpg', 0.341)
2 º : ('20160905_074151_000.jpg', 0.333)
3 º : ('20160905_074327_000.jpg', 0.326)
4 º : ('20160905_075317_000.jpg', 0.079)
5 º : ('20160905_075349_000.jpg', 0.067)
6 º : ('20160905_075213_000.jpg', 0.063)
7 º : ('20160905_075109_000.jpg', 0.062)
8 º : ('20160905_075141_000.jpg', 0.059)
9 º : ('20160829_075953_000.jpg', 0.054)
10 º : ('20160905_075245_000.jpg', 0.054)
```

Figura 9: Resultados do tópico LSC27 obtidos pelo algoritmo.

```
IMAGE MAX SCORE

1 º :
2 º : ('B00001412_21I6X0_20180512_214030E.JPG', 0.151)
3 º : ('B00001435_21I6X0_20180512_214806E.JPG', 0.15)
4 º : ('20160827_080625_000.jpg', 0.143)
5 º : ('B00001411_21I6X0_20180512_214022E.JPG', 0.08)
6 º : ('20160820_171534_000.jpg', 0.069)
7 º : ('B00000395_21I6X0_20180512_191403E.JPG', 0.068)
8 º : ('B00007444_21I6X0_20180505_204056E.JPG', 0.052)
9 º : ('B00007442_21I6X0_20180505_204008E.JPG', 0.051)
10 º : ('B00004786_21I6X0_20180526_195952E.JPG', 0.0)
```

Figura 10: Resultados do tópico LSC28 obtidos pelo algoritmo.

```
IMAGE MAX SCORE

1 º : ('20160910_122846_000.jpg', 0.231)
2 º : ('b00000637_21i6bq_20150228_122842e.jpg', 0.15)
3 º : ('B00004087_21I6X0_20180526_130620E.JPG', 0.144)
4 º : ('B00004097_21I6X0_20180526_130954E.JPG', 0.14)
5 º : ('B00004099_21I6X0_20180526_131040E.JPG', 0.138)
6 º : ('20160820_131734_000.jpg', 0.137)
7 º : ('B00004090_21I6X0_20180526_130729E.JPG', 0.135)
8 º : ('20160820_120434_000.jpg', 0.127)
9 º : ('b00000686_21i6bq_20150228_130105e.jpg', 0.117)
10 º : ('20160910_122918_000.jpg', 0.108)
11 º : ('20160903_124457_000.jpg', 0.105)
12 º : ('20160903_124353_000.jpg', 0.101)
13 º : ('20160924_121757_000.jpg', 0.082)
```

Figura 11: Resultados do tópico LSC30 obtidos pelo algoritmo.

```
IMAGE MAX SCORE

1 º : ('20160905_105347_000.jpg', 0.307)
2 º : ('20160905_112158_000.jpg', 0.299)
3 º : ('20160905_112406_000.jpg', 0.286)
4 º : ('20160905_105744_000.jpg', 0.277)
5 º : ('20160905_111032_000.jpg', 0.274)
6 º : ('20160829_115342_000.jpg', 0.273)
7 º : ('20160905_111104_000.jpg', 0.27)
8 º : ('20160905_110720_000.jpg', 0.267)
9 º : ('20160905_110304_000.jpg', 0.266)
10 º : ('20160829_114646_000.jpg', 0.264)
11 º : ('20160829_100330_000.jpg', 0.103)
12 º : ('20160829_100434_000.jpg', 0.102)
13 º : ('20160905_110928_000.jpg', 0.1)
14 º : ('20160829_083708_000.jpg', 0.1)
15 º : ('20160829_095810_000.jpg', 0.1)
16 º : ('20160829_094644_000.jpg', 0.099)
17 º : ('20160829_095653_000.jpg', 0.099)
18 º : ('20160829_101202_000.jpg', 0.099)
19 º : ('20160905_084456_000.jpg', 0.099)
20 º : ('20160905_094449_000.jpg', 0.099)
21 º : ('20160905_094833_000.jpg', 0.099)
22 º : ('20160905_110408_000.jpg', 0.099)
```

Figura 12: Resultados do tópico LSC36 obtidos pelo algoritmo.

```
IMAGE MAX SCORE

1 º : ('20160908_181348_000.jpg', 0.266)
2 º : ('20160908_181452_000.jpg', 0.253)
3 º : ('20160908_181316_000.jpg', 0.245)
4 º : ('20160908_174933_000.jpg', 0.244)
5 º : ('20160908_180203_000.jpg', 0.238)
6 º : ('20160908_173821_000.jpg', 0.237)
7 º : ('20160908_173319_000.jpg', 0.237)
8 º : ('20160908_180900_000.jpg', 0.233)
9 º : ('20160908_181212_000.jpg', 0.233)
10 º : ('20160908_180131_000.jpg', 0.233)
11 º : ('b00001143_21i6bq_20150312_200521e.jpg', 0.114)
12 º : ('b00001097_21i6bq_20150312_191337e.jpg', 0.108)
13 º : ('b00001146_21i6bq_20150312_200711e.jpg', 0.108)
14 º : ('20160908_181420_000.jpg', 0.102)
15 º : ('20160908_181140_000.jpg', 0.094)
16 º : ('b00001141_21i6bq_20150312_200217e.jpg', 0.09)
17 º : ('b00001142_21i6bq_20150312_200409e.jpg', 0.088)
18 º : ('b00001145_21i6bq_20150312_200631e.jpg', 0.086)
19 º : ('b00001147_21i6bq_20150312_200749e.jpg', 0.085)
```

Figura 13: Resultados do tópico LSC39 obtidos pelo algoritmo.

```
20160813_131130_000.jpg, 0.67
20160820_131734_000.jpg, 0.44
```

Figura 14: Resultados do tópico LSC26 obtidos pelo LoggyApp.

```
20160905_051627_000.jpg, 0.59
20160905_051731_000.jpg, 0.68
20160905_051835_000.jpg, 0.58
20160905_052603_000.jpg, 0.62
B00009892_21I6X0_20180507_102059E.JPG, 0.65
```

Figura 15: Resultados do tópico LSC27 obtidos pelo LoggyApp.

```
20160813_123734_000.jpg, 0.47
20160813_125809_000.jpg, 0.41
20160813_140659_000.jpg, 0.41
20160813_142908_000.jpg, 0.44
20160813_143042_000.jpg, 0.42
20160813_143611_000.jpg, 0.44
20160813_144639_000.jpg, 0.45
20160813_144814_000.jpg, 0.41
20160813_145035_000.jpg, 0.45
20160813_145256_000.jpg, 0.46
20160813_145517_000.jpg, 0.41
20160813_153059_000.jpg, 0.43
20160813_154518_000.jpg, 0.44
20160813_154826_000.jpg, 0.41
20160820_114925_000.jpg, 0.40
20160820_170838_000.jpg, 0.40
20160820_170806_000.jpg, 0.40
20160820_170910_000.jpg, 0.42
20160820_171014_000.jpg, 0.42
20160903_094036_000.jpg, 0.41
20160903_094153_000.jpg, 0.41
20160903_094329_000.jpg, 0.44
20160903_094433_000.jpg, 0.40
20160903_094401_000.jpg, 0.40
20160910_110850_000.jpg, 0.43
20160910_110954_000.jpg, 0.41
20160910_111026_000.jpg, 0.43
20160910_111058_000.jpg, 0.42
20160910_115453_000.jpg, 0.44
20160910_115557_000.jpg, 0.44
20160910_115629_000.jpg, 0.45
20160910_115734_000.jpg, 0.47
20160910_115701_000.jpg, 0.46
20160910_115838_000.jpg, 0.46
20160910_115806_000.jpg, 0.47
20160910_115942_000.jpg, 0.47
20160910_115910_000.jpg, 0.42
20160910_120046_000.jpg, 0.45
20160910_120014_000.jpg, 0.46
20160910_120118_000.jpg, 0.45
20160910_120150_000.jpg, 0.47
20160910_120254_000.jpg, 0.44
20160910_120222_000.jpg, 0.43
20160910_120326_000.jpg, 0.46
20160910_120358_000.jpg, 0.45
20160910_120607_000.jpg, 0.43
20160910_120639_000.jpg, 0.40
20160910_120743_000.jpg, 0.42
20160910_120847_000.jpg, 0.44
20160910_120815_000.jpg, 0.44
20160910_120919_000.jpg, 0.46
20160910_120951_000.jpg, 0.46
```

Figura 16: Resultados do tópico LSC28 obtidos pelo LoggyApp.

```
20160910_122846_000.jpg, 0.28
20160917_153444_000.jpg, 0.65
20160917_153549_000.jpg, 0.39
20160917_173015_000.jpg, 0.28
```

Figura 17: Resultados do tópico LSC30 obtidos pelo LoggyApp.

```
20160815_074827_000.jpg, 0.34
20160815_081223_000.jpg, 0.46
20160815_082631_000.jpg, 0.48
20160815_083608_000.jpg, 0.49
20160822_081533_000.jpg, 0.38
20160822_081501_000.jpg, 0.39
20160905_080149_000.jpg, 0.31
20160905_080429_000.jpg, 0.32
20160912_083007_000.jpg, 0.34
20160912_084537_000.jpg, 0.47
20160912_092737_000.jpg, 0.35
B00009645_21I6X0_20180507_084127E.JPG, 0.35
B00009649_21I6X0_20180507_084306E.JPG, 0.31
B00009661_21I6X0_20180507_084732E.JPG, 0.34
B00009795_21I6X0_20180507_094204E.JPG, 0.31
B00009802_21I6X0_20180507_094447E.JPG, 0.35
B00009801_21I6X0_20180507_094425E.JPG, 0.38
B00009804_21I6X0_20180507_094534E.JPG, 0.32
B00000833_21I6X0_20180514_075203E.JPG, 0.34
B00002973_21I6X0_20180521_084332E.JPG, 0.41
B00006808_21I6X0_20180528_075053E.JPG, 0.38
B00006817_21I6X0_20180528_075424E.JPG, 0.33
B00006820_21I6X0_20180528_075533E.JPG, 0.44
B00006822_21I6X0_20180528_075619E.JPG, 0.40
B00006840_21I6X0_20180528_080331E.JPG, 0.39
B00006843_21I6X0_20180528_080448E.JPG, 0.39
B00006844_21I6X0_20180528_080512E.JPG, 0.36
```

Figura 18: Resultados do tópico LSC36 obtidos pelo LoggyApp.

```
20160908_180131_000.jpg, 0.55
20160908_180203_000.jpg, 0.44
20160908_180307_000.jpg, 0.48
20160908_180339_000.jpg, 0.51
20160908_180443_000.jpg, 0.47
20160908_180411_000.jpg, 0.50
20160908_180516_000.jpg, 0.48
20160908_180652_000.jpg, 0.42
20160908_180620_000.jpg, 0.47
20160908_180900_000.jpg, 0.42
20160908_180932_000.jpg, 0.38
20160908_181036_000.jpg, 0.44
20160908_181108_000.jpg, 0.51
20160908_181140_000.jpg, 0.51
20160908_181212_000.jpg, 0.47
20160908_181348_000.jpg, 0.49
20160908_181316_000.jpg, 0.51
20160908_181420_000.jpg, 0.52
20160908_181452_000.jpg, 0.50
```

Figura 19: Resultados do tópico LSC39 obtidos pelo LoggyApp.