



Universidade do Minho
Escola de Engenharia

UNIVERSIDADE DO MINHO

LABORATÓRIO EM ENGENHARIA
INFORMÁTICA

Projeto LEI

Trabalho Realizado por:
Guilherme Palumbo, PG42832
João Silva, PG42834

Grupo:
105

Orientado por:
José Machado
Regina Sousa

28 de junho de 2021

Resumo

Nos dias de hoje, é cada vez mais importante a informatização da informação, uma vez que permite uma maior eficiência nos processos, uma entrega mais personalizada ao cliente e controle com maior qualidade da informação. Para além disto, temos também as exigências do mercado do trabalho que são cada vez maiores, pelo que é necessário uma resposta breve e com a melhor qualidade possível. A pandemia que assombrou as nossas vidas ao longo dos últimos meses, veio comprovar isso mesmo e acelerar um pouco o processo de transição digital. Neste sentido, neste trabalho procuramos com base num conjunto de dados retirado do *Kaggle* informatizar a informação do mesmo. Para isso começamos por importar o conjunto de dados para uma base de dados NoSQL (no caso MongoDB) uma vez que é capaz de processar grandes conjuntos de dados não estruturados e em constante mudança. De seguida, desenvolvemos uma plataforma online onde é possível visualizar o conjunto de dados de forma organizada.

Por último, e após ter feito uma análise detalhada ao conjunto de dados, preparamos os mesmos e desenvolvemos um algoritmo de *Text Mining* sobre a coluna *abstract*. Para além do algoritmo, apresentamos também uma outra técnica onde com base em algumas questões conseguimos encontrar quais os artigos que mais se identificam com as mesmas. Para realizar estas tarefas utilizamos a ferramenta *Colab* e algumas bibliotecas do *python* (por exemplo *nlkt*).

Palavras Chave: MongoDB, Desenvolvimento Web, Node.js, Text Mining, Covid-19.

Conteúdo

1	Introdução	5
1.1	Identificação do Projeto e Objectivos	5
1.2	Metodologia de Trabalho	5
1.3	Estrutura do Relatório	7
2	Descrição e Análise de Dados	8
3	Tarefa 1 - Base de Dados	13
4	Tarefa 2 - Plataforma Online	15
5	Tarefa 3 - Text Mining	23
6	Conclusão	30
6.1	Trabalho Futuro	30

Lista de Figuras

1	Metodologia CRISP-DM.	6
2	Valores Duplicados.	9
3	Valores em Falta.	9
4	Representação Gráfica dos Valores em Falta.	9
5	Número de Ocorrências do atributo <i>source_x</i>	10
6	Número de Ocorrências do atributo <i>license</i>	10
7	Top 15 jornais com mais autores e títulos.	11
8	Licenças com mais títulos e autores e Top 15 de autores com mais títulos.	11
9	Top 15 dos jornais e autores com mais títulos.	12
10	Mongod.	13
11	MongoImport.	13
12	Bases de Dados presentes no host 127.0.0.1:27017.	14
13	Base de Dados LEI2020.	14
14	Documents dentro da collection articles.	14
15	Conexão da app ao MongoDB.	15
16	Diagrama da relação Model-View-Controller.	16
17	Exemplo do Model articles.	16
18	Exemplo de uma Query em mongoDB que retorna todos os dados.	17
19	Exemplo de uma Query em mongoDB que retorna um dado por id.	17

20	Script das routes com os GETs necessários	18
21	View de todos os artigos	18
22	View de um artigo em específico	19
23	Script do sistema de pesquisa.	20
24	Controller do sistema de pesquisa.	20
25	Route do sistema de pesquisa.	20
26	Página com todos os artigos.	21
27	Resultado de um search por "trans".	21
28	Resultado de um search pela data "2001-07-04".	21
29	Resultado de um search por "Respir".	22
30	Resultado de um search pelo autor "vliet".	22
31	Página individual com apenas 1 artigo.	22
32	Remover algumas colunas.	23
33	Remover valores em falta.	24
34	Remover pontuação.	24
35	Nuvem de Palavras.	25
36	Remover <i>stopwords</i> e tokenizar os dados.	25
37	Representação do Modelo LDA.	26
38	Modelo LDA desenvolvido.	27
39	Visualização dos tópicos para interpretação.	28
40	Selecionar os artigos com as palavras <i>covid</i> , <i>-cov-2</i> , <i>cov2</i> e <i>ncov</i>	29
41	Resultados obtidos para algumas questões.	29

1 Introdução

1.1 Identificação do Projeto e Objectivos

No âmbito da Unidade Curricular Laboratório em Engenharia Informática foi nos proposto a realização de um trabalho prático que consiste na concretização de uma plataforma online e um modelo Machine Learning que permita realizar *Text Mining* ao dataset, sendo que para isso devemos utilizar o seguinte dataset referente ao *COVID 19 Open Research Data Challenge Code* encontrado no kaggle: <https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge>. Os principais objetivos deste trabalho são:

1. Montar o *dataset* numa máquina;
2. Fazer *display* do dataset numa plataforma online onde seja possível pesquisar por palavras chave, autores, etc;
3. Fazer *Text Mining* aos *abstracts* e apresentar os resultados.

1.2 Metodologia de Trabalho

Para a realização deste trabalho, utilizamos como metodologia de trabalho, a Metodologia CRISP-DM, uma vez que é uma metodologia que se baseia na experiência prática de como as pessoas desenvolvem os projetos de data mining [1, 2].

A metodologia CRISP-DM é constituída por 6 fases, sendo estas:

1. **Estudo do negócio** - identificar e compreender o problema da empresa que precisa de ser resolvido. É preciso compreender o negócio de acordo com seus objetivos e perspectivas e, por consequência, definir quais são as suas necessidades. Em suma, é converter os objetivos e requisitos de negócio num problema de mineração de dados;
2. **Estudo dos dados** – após entender o negócio e definir os objetivos, é imprescindível conhecer os dados e identificar quais são os mais relevantes para a solução do problema em questão. É necessário verificar e organizar todos os dados disponíveis e que são indispensáveis para solucionar o problema em questão. Nesta etapa refere-se como os dados foram adquiridos e descrevem-se informações relevantes, como o seu formato e o conjunto de valores que podem tomar, de forma a identificar e compreender a informação contida neles;
3. **Preparação dos dados** – é onde se realizado um conjunto de tarefas de inspeção e preparação dos dados com o intuito de se obterem os dados finais, para proceder à criação e validação dos modelos. Posto isto, podem-se executar ações para obter dados mais limpos, como: seleccionar, combinar, transformar e substituir valores em falta;

4. **Modelação** – é onde se seleciona e aplica os modelos tendo em conta os objetivos definidos. Nesta etapa, aplicam-se os algoritmos, capazes de produzir resultados satisfatórios sobre o conjunto de dados preparado na fase anterior;
5. **Avaliação** – é onde se avalia o desempenho dos modelos desenvolvidos na fase anterior, tendo em conta as métricas de avaliação pré-definidas. De certa forma, é onde percebemos se os objetivos foram alcançados;
6. **Implementação** - em que se faz a aplicação do modelo no processo de tomada de decisão.

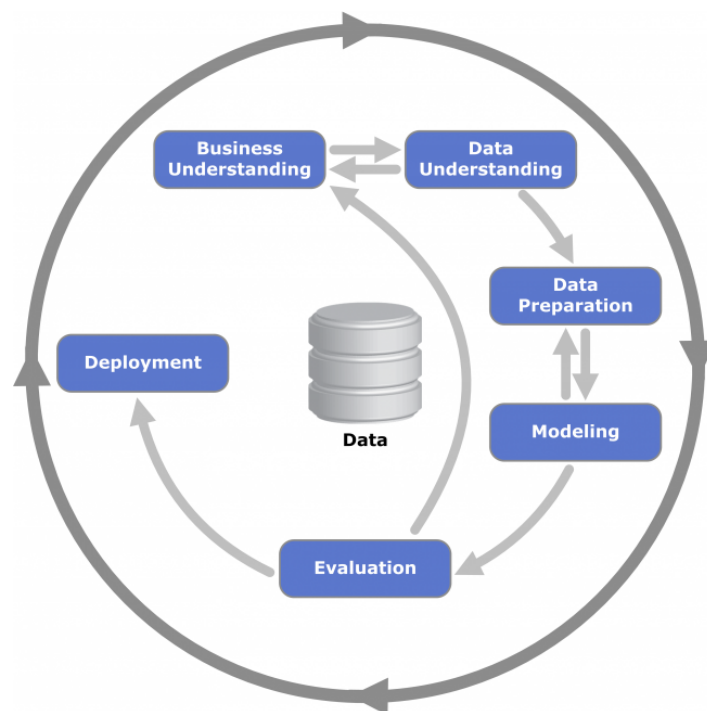


Figura 1: Metodologia CRISP-DM.

1.3 Estrutura do Relatório

Para além deste capítulo, este relatório está organizado em cinco capítulos, são eles:

Capítulo 2: Descrição e Análise de Dados – Capítulo onde fazemos uma análise do *dataset* inicial para compreendermos com que tipo de dados estamos a trabalhar.

Capítulo 3: Tarefa 1 - Base de Dados – Neste capítulo carregamos o ficheiro do dataset para a base de dados *NoSQL MongoDB* por ser a mais adequada para a realização das operações futuras.

Capítulo 4: Tarefa 2 - Plataforma online – Neste capítulo desenvolvemos uma plataforma online onde é possível a visualização de todos os artigos presentes no *dataset* e um motor de busca que permite pesquisar os artigos por palavras-chave, autores, títulos, etc.

Capítulo 5: Tarefa 3 - TextMining – Neste capítulo com o auxílio do Colab desenvolvemos técnicas/algoritmos *Text Mining* aos *abstracts* e apresentamos os seus resultados.

Capítulo 6: Conclusão – Capítulo onde fazemos algumas conclusões sobre resultados obtidos e trabalho futuro.

Capítulo 7: Referências Bibliográficas – Neste capítulo estão referenciadas, de acordo com as normas APA, todas as fontes utilizadas.

2 Descrição e Análise de Dados

O conjunto de dados que utilizamos para atingir os objetivos deste projeto foi o *COVID-19 Open Research Dataset Challenge (CORD-19)*, retirado de: <https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge?select=metadata.csv>. Este conjunto de dados é constituído por mais de 500 mil artigos académicos sobre o *COVID-19*, *SARS-Cov-2* e *coronarívus*. Sendo o *coronavírus* um tema tão atual e onde todas as descobertas são importantes, o objetivo deste conjunto de dados é com base no processamento de linguagem natural e outras técnicas de inteligência artificial (IA), gerar novos insights em apoio à luta da pandemia que nos assombra. Para isso temos mais de 500 mil registos caracterizados por 19 atributos, sendo estes:

1. *cord_uid*
2. *sha*
3. *source_x*: fonte do artigo;
4. *title*: título do artigo;
5. *doi*
6. *pmcid*: identificador único do *pmc*
7. *pubmed_id*: identificador único do *pubmed*;
8. *license*: licença do artigo;
9. *abstract*: resumo do artigo;
10. *publish_time*: data de publicação do artigo;
11. *authors*: autores do artigo;
12. *journal*: jornal que publicou o artigo;
13. *mag_id*: identificador único do *mag*;
14. *who_convidence_id*: identificador único do *who_convidence*;
15. *arxiv_id*: identificador único do *arxiv*;
16. *pdf_json_files*: url para ficheiros no formato pdf e json;
17. *pmc_json_files*: url para ficheiros no formato pmc e json;
18. *url*: url onde o artigo pode ser acedido;
19. *s2_id*: identificador único do *s2*.

Uma vez apresentada uma breve descrição do conjunto de dados, de seguida efetuamos uma análise detalhada ao nosso conjunto de dados. Começamos por verificar se existiam valores duplicados (Figura 2) e valores em falta (Figura 3 e Figura 4).

```
[36] #duplicated values
df_inicial.duplicated().sum()

0
```

Figura 2: Valores Duplicados.

```
#missing values
df_inicial.isnull().sum()

cord_uid      0
sha           399672
source_x      0
title         302
doi           278317
pmcid         390344
pubmed_id     322954
license       0
abstract      160399
publish_time   219
authors       15622
journal       39062
mag_id        599616
who_covidence_id 342106
arxiv_id       591775
pdf_json_files 399672
pmc_json_files 437663
url           255863
s2_id         52771
dtype: int64
```

Figura 3: Valores em Falta.

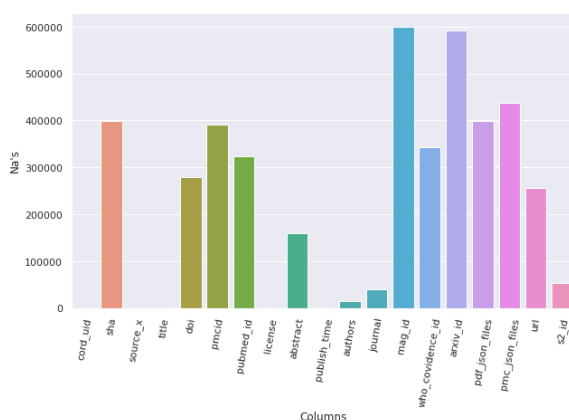


Figura 4: Representação Gráfica dos Valores em Falta.

Seguidamente, e tendo em conta a grandeza do conjunto de dados, para alguns

atributos fizemos uma análise individual, onde contamos o número de ocorrências para cada instância do atributo (Figura 5 e Figura 6).

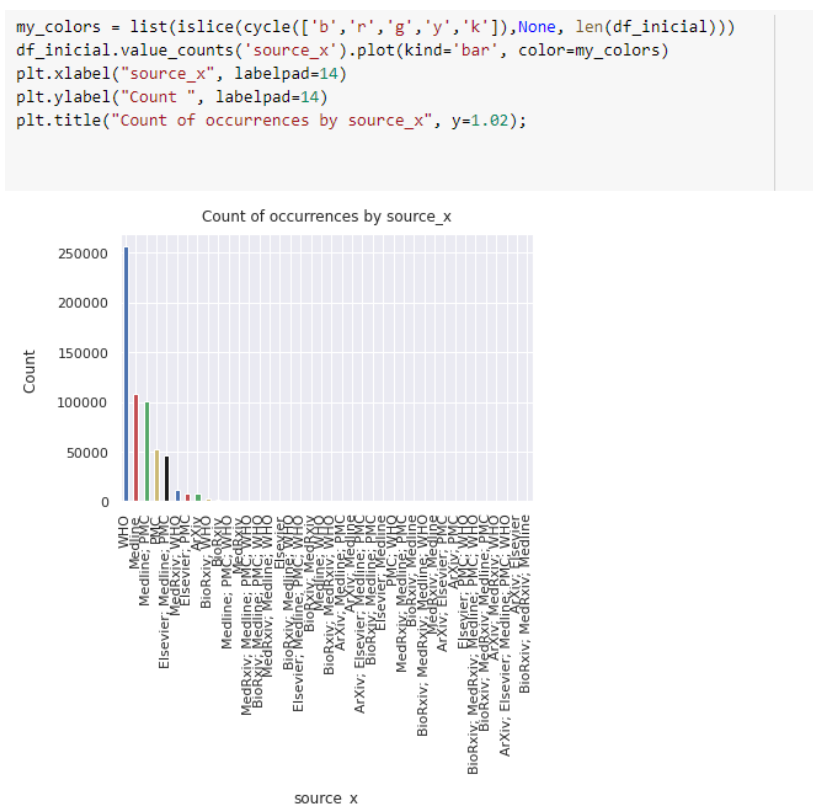


Figura 5: Número de Ocorrências do atributo *source_x*.

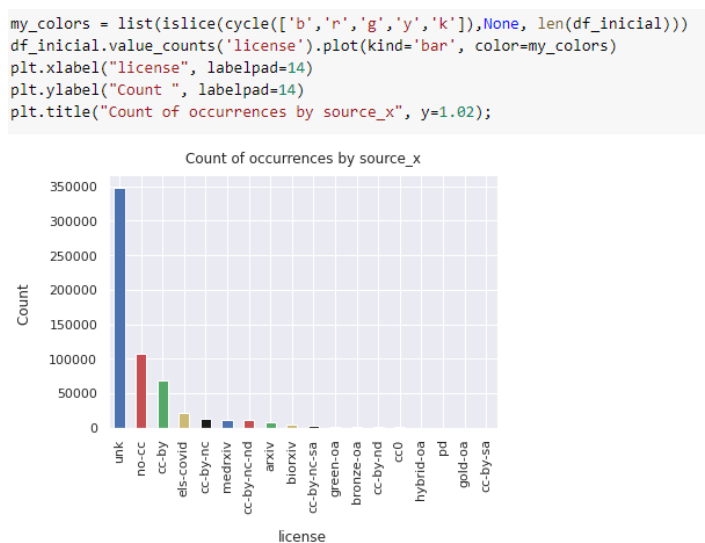


Figura 6: Número de Ocorrências do atributo *license*.

De seguida, através de alguns *groupby*, realizamos uma análise "composta", ou seja, pegamos em mais do que uma variável (na maioria das vezes duas variáveis) e contamos quantas ocorrências haviam tendo em conta esses atributos. Por exemplo, quais os jornais com mais autores, quais os autores com mais títulos, entre outros (Figura 7a, Figura 7b, Figura 8a, Figura 8b e Figura 9).

authors	journal
bioRxiv	5589
PLoS One	5553
BMJ	4888
Sci Rep	2885
Int J Environ Res Public Health	2802
Lancet	2178
Nature	2124
Journal of virology	1938
Cureus	1886
JAMA	1876
Viruses	1807
Int J Infect Dis	1757
Science	1589
Int. j. environ. res. public health (Online)	1587
Front Immunol	1510

title	journal
bioRxiv	5589
PLoS One	5575
BMJ	5060
Sci Rep	2887
Int J Environ Res Public Health	2803
Nature	2413
Lancet	2369
Journal of virology	1940
JAMA	1924
Cureus	1887
Viruses	1807
Int J Infect Dis	1764
Science	1618
Int. j. environ. res. public health (Online)	1587
Front Immunol	1511

(a) Top 15 dos jornais com mais autores. (b) Top 15 dos jornais com mais títulos.


Figura 7: Top 15 jornais com mais autores e títulos.

license	title	authors
unk	348445	341564
no-cc	107867	101167
cc-by	68297	67784
els-covid	20761	19914
cc-by-nc	12547	12442
medrxiv	11638	11638
cc-by-nc-nd	11119	10973
arxiv	7717	7717
biornxiv	4649	4649
cc-by-nc-sa	3273	3144
green-oa	858	854
bronze-oa	657	633
cc-by-nd	596	636
cc0	495	491
hybrid-oa	153	152
pd	126	122
gold-oa	87	86
cc-by-sa	29	28

authors	title
Anonymous,	2046
Mahase, Elisabeth	465
Iacobucci, Gareth	365
Rimmer, Abi	272
Prevention, Centers for Disease Control and	207
Wise, Jacqui	181
Manus, Jean-Marie	174
Organization, World Health	162
Dyer, Owen	143
Joob, Beuy; Wiwanitkit, Viroj	117
Kuehn, Bridget M	110
Dyer, Clare	109
Saúde, Organização Pan-Americana da	107
Kow, Chia Siang; Hasan, Syed Shahzad	105
Canady, Valerie A.	103

(a) Licenças com mais títulos e autores. (b) Top 15 de autores com mais títulos.

Figura 8: Licenças com mais títulos e autores e Top 15 de autores com mais títulos.



	journal	authors	title
BMJ		Mahase, Elisabeth	449
		Iacobucci, Gareth	355
		Rimmer, Abi	263
		Wise, Jacqui	177
		Dyer, Owen	138
Rev Francoph Lab		Manus, Jean-Marie	122
JAMA		Kuehn, Bridget M	104
BMJ		Dyer, Clare	104
		Tanne, Janice Hopkins	102
JAMA		Rubin, Rita	92
BMJ		Torjesen, Ingrid	89
		Oliver, David	88
Nature		Mallapaty, Smriti	84
BMJ		Griffin, Shaun	79
Science		Cohen, Jon	76

Figura 9: Top 15 dos jornais e autores com mais títulos.

Após a análise detalhada do nosso conjunto de dados conseguimos obter algumas informações relevantes, nomeadamente que:

- não temos valores duplicados;
- temos bastantes valores em falta, chegando mesmo algumas colunas a ter a totalidade dos valores em falta (por exemplo *mag_id*, *arxiv_id*);
- devido à enorme quantidade de valores em falta algumas colunas tornam-se inutilizáveis;
- as *source_x* mais representativas são *WHO*, *Medline* e *MedlinePMC*;
- as licenças mais representativas são *unlk*, *no_cc* e *cc_by*;
- os jornais com mais autores e títulos são *bioRxiv*, *PloS One* e *BMJ*;
- os autores com mais títulos são *Mahase, Elisabeth*, *Iacobucci, Gareth* e *Rimmer, Abi*;
- alguns títulos estão registados como anónimos;
- as licenças com mais títulos e autores são *unk*, *no-cc* e *cc-by*;
- os autores com mais títulos pertencem ao jornal *BMJ*.

3 Tarefa 1 - Base de Dados

Nesta primeira fase temos como objetivo importar o dataset para uma base de dados. Neste caso a melhor opção para resolver este problema seria utilizar uma base de dados NoSQL (não relacional) pois é capaz de processar grandes volumes de dados não estruturados e em constante mudança [3].

Para tal, utilizamos a ferramenta MongoDB, uma base de dados NoSQL distribuída, *document-based*, criada para desenvolvedores de aplicações modernas e para a era da *cloud* (nuvem). Uma das vantagens da base de dados MongoDB é que é uma base de dados de documentos o que significa que ela armazena dados em documentos do tipo JSON. Visto que JSON é um tipo de ficheiro muito simples isso leva a que ao trabalhar com o MongoDB a maneira de trabalhar com dados seja mais natural e produtiva. Um dos benefícios desta ferramenta é que oferece suporte a matrizes (*arrays*) e objetos aninhados (*nested objects*) como valores (*values*). Por último também permite esquemas (*schemas*) flexíveis e dinâmicos [4].

De forma a importarmos o nosso ficheiro CSV para a base de dados MongoDB, primeiro iniciamos o *mongod* que é o processo principal daemon para o sistema MongoDB. Ele lida com solicitações de dados, gerencia o acesso aos dados e executa operações de gerenciamento em segundo plano (Figura 10) [5]. Depois de iniciado o processo, através do terminal no diretório apropriado, fizemos um *mongoimport*, uma funcionalidade do mongo, criamos um novo repositório com o nome "LEI2020", uma nova coleção com o nome de "articles", identificamos que tipo de documento se tratava (csv) e indicamos o nome do ficheiro (Figura 11).

```
C:\Users\Utilizador>mongod
{"t":{"$date":"2021-06-22T18:11:01.430+01:00"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"main", "msg":"Automaticall
disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2021-06-22T18:11:01.819+01:00"},"s":"W",  "c":"ASIO",    "id":22601,   "ctx":"main", "msg":"No Transport
layer configured during NetworkInterface startup"}
{"t":{"$date":"2021-06-22T18:11:01.819+01:00"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main", "msg":"Implicit TCP
FastOpen in use."}
{"t":{"$date":"2021-06-22T18:11:01.821+01:00"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx":"initandlisten", "msg":"Mon
gDB starting", "attr":{"pid":9332, "port":27017, "dbPath":"C:/data/db/", "architecture":"64-bit", "host":"DESKTOP-OUAC1U7"}}
{"t":{"$date":"2021-06-22T18:11:01.821+01:00"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten", "msg":"Tar
get operating system minimum version", "attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2021-06-22T18:11:01.821+01:00"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten", "msg":"Bui
ld Info", "attr":{"buildInfo":{"version":"4.4.2", "gitVersion":"15e73dc5738d2278b688f8929aee605fe4279b0e", "modules":[], "al
locator":"tcmalloc", "environment":{"distmod":"windows", "distarch":"x86_64", "target_arch":"x86_64"}}}}
{"t":{"$date":"2021-06-22T18:11:01.821+01:00"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten", "msg":"Ope
rating System", "attr":{"os":{"name":"Microsoft Windows 10", "version":"10.0 (build 19042)"}}
{"t":{"$date":"2021-06-22T18:11:01.821+01:00"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten", "msg":"Opt
ions set by command line", "attr":{"options":{}}}
{"t":{"$date":"2021-06-22T18:11:01.823+01:00"},"s":"I",  "c":"STORAGE",  "id":22270,   "ctx":"initandlisten", "msg":"Sto
rage engine to use detected by data files", "attr":{"dbpath":"C:/data/db/", "storageEngine":"wiredTiger"}}
{"t":{"$date":"2021-06-22T18:11:01.823+01:00"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":"initandlisten", "msg":"Ope
ning WiredTiger", "attr":{"config":"create cache size=5576M session max=33000 eviction=(threads_min=4 threads_max=4) confi
```

Figura 10: Mongod.

```
PS C:\Users\Utilizador\Desktop\LEI> cd .\data\
PS C:\Users\Utilizador\Desktop\LEI\data> mongoimport --db=LEI2020 --collection=articles --type=csv --file=metadata.csv
```

Figura 11: MongoImport.

Nas figuras 12, 13 e 14 podemos ver que o dataset foi efetivamente importado para o MongoDB, concluindo assim a Tarefa número 1.

Database Name ^	Storage Size	Collections	Indexes	
LEI2020	544.1MB	1	1	
TP_DAW	64.0KB	2	4	
admin	32.0KB	0	1	
config	12.0KB	0	2	
local	44.0KB	1	1	

Figura 12: Bases de Dados presentes no host 127.0.0.1:27017.

Collection Name ^	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
articles	599,616	1.7 KB	971.2 MB	1	5.5 MB	

Figura 13: Base de Dados LEI2020.

LEI2020.articles

DOCUMENTS 599.6k TOTAL SIZE 971.2MB AVG. SIZE 1.7KB INDEXES 1 TOTAL SIZE 5.5MB AVG. SIZE 5.5MB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } **OPTIONS** **FIND** **RESET** **...**

ADD DATA **VIEW** **...** Displaying documents 1 - 20 of 599616 **C REFRESH**

```

_id: ObjectId("60c8bdfadb82b2ba1235c93f")
cord_uid: "ug7v899j"
sha: "d1aafb70c866a2068b02786f8929fd9c90897fb"
source_x: "PMC"
title: "Clinical features of culture-proven Mycoplasma pneumoniae infections a..."
doi: "10.1186/1471-2334-1-6"
pmcid: "PMC35282"
pubmed_id: 11472636
license: "no-cc"
abstract: "OBJECTIVE: This retrospective chart review describes the epidemiology ..."
publish_time: "2001-07-04"
authors: "Madani, Tariq A; Al-Ghamdi, Aisha A"
journal: "BMC Infect Dis"
mag_id: ""
who_covidence_id: ""
arxiv_id: ""
pdf_json_files: "document_parses/pdf_json/d1aafb70c866a2068b02786f8929fd9c90897fb.json"
pmc_json_files: "document_parses/pmc_json/PMC35282.xml.json"
url: "https://www.ncbi.nlm.nih.gov/pmc/articles/PMC35282/"
s2_id: ""

_id: ObjectId("60c8bdfadb82b2ba1235c940")
cord_uid: "827nu04a"
sha: "cb0567720c2143a66d737eb0a2f63f2dce2e5a7d"
source_x: "PMC"
title: "Nitric oxide: a pro-inflammatory mediator in lung disease?"
doi: "10.1186/rr14"
pmcid: "PMC59543"
pubmed_id: 11667967
license: "no-cc"
abstract: "Inflammatory diseases of the respiratory tract are commonly associated..."

```

Figura 14: Documents dentro da collection articles.

4 Tarefa 2 - Plataforma Online

A segunda Tarefa consiste em desenvolver uma plataforma online que permita fazer display do dataset e consequentemente seja possível fazer pesquisas por palavras-chave, autores, etc.

Para resolver a tarefa utilizamos o `node.js` que é um tempo de execução de *JavaScript* construído no motor V8 *JavaScript* do *Chrome* [6]. Utilizamos *express.js* que é uma *framework* de aplicações web *node.js* mínima e flexível que fornece um conjunto robusto de recursos para aplicativos da web e móveis [7]. Para o frontend utilizamos *pug.js*, um *template engine* para *node.js* que permitirá injetar dados e, em seguida, produzir HTML [8].

Começamos então por gerar uma `app.js` utilizando `express` e consequentemente configuramos a mesma para ligar diretamente à base de dados MongoDB (Figura 15).

```
//Set up default mongoose connection
const mongoose = 'mongodb://127.0.0.1/LEI2020';
mongoose.connect(mongoose, { useNewUrlParser: true, useUnifiedTopology: true, useCreateIndex: true });

//Get the default connection
const db = mongoose.connection;

//Bind connection to error event (to get notification of connection errors)
db.on('error', console.error.bind(console, 'MongoDB connection error...'));
db.once('open', function () {
  console.log("Successful MongoDB connection ...")
});
```

Figura 15: Conexão da app ao MongoDB.

Para efeitos de teste estaremos a utilizar um localhost na port 3000 e visto que o dataset possui mais de 50 mil registos, estaremos a utilizar apenas 100.

Na fase efetiva de desenvolvimento precisamos de definir um padrão de arquitetura de software. Visto que a *framework express* já possui um padrão, estaremos esse padrão que é o MVC, Model-View-Controller.

O padrão de arquitetura de software é constituído por 3 fases, uma fase de Model (dados), View (*layout*) e Controller como o próprio nome indica. Desta forma, alterações feitas no *layout* não afectam a manipulação de dados, e estes poderão ser reorganizados sem alterar o *layout*. O model-view-controller resolve este problema através da separação das tarefas de acesso aos dados e lógica de negócio, lógica de apresentação e de interação com o utilizador, introduzindo um componente entre os dois: o Controller [9] (Figura 16).

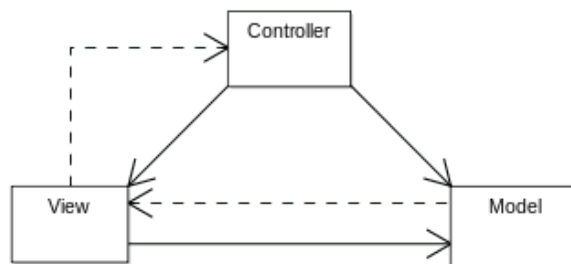


Figura 16: Diagrama da relação Model-View-Controller.

O Model é a representação "domínio" específica da informação em que a aplicação opera. No nosso caso temos um Model dos artigos onde iremos especificar todos os atributos presentes dentro do Model (seria o equivalente a uma tabela numa base de dados relacional). O nosso Model artigos é um *schema* que possuirá todos os atributos presentes no nosso dataset tais como o titulo, data, pmcid, entre outros (exemplo na figura 17).

```

var articleSchema = new mongoose.Schema({
  cord_uid: {
    type: String
  },
  sha: {
    type: String
  },
  source_x: {
    type: String
  },
  title: {
    type: String
  },
  doi: {
    type: String
  },
  pmcid: {
    type: String
  },
  . . . . .
})
  
```

Figura 17: Exemplo do Model articles.

De seguida iremos definir o Controller que tem o papel de intermediário entre o Model e a View, basicamente ele vai processar e responder a eventos, geralmente ações do utilizador, e pode invocar alterações no Model. É lá que é feita a validação dos dados e também é onde os valores postos pelos usuários são filtrados [9].

É no Controller que se faz efetivamente as APIs à base de dados, utilizando a linguagem de query do mongoDB, dado que é uma linguagem de consulta rica e expressiva que

permite filtrar e classificar por qualquer campo, não importa o quão aninhado ele possa estar em um documento. Esse tipo de query também dá suporte a agregações e outros casos de uso modernos, como pesquisa baseada em localização geográfica, pesquisa de gráfico e pesquisa de texto e as próprias queries são JSON e, portanto, facilmente combináveis. Por último facilita o processo pois não é mais necessário concatenar *strings* para gerar consultas SQL dinamicamente [4].

Posto isto, desenvolvemos algumas queries tais como uma query em que o nosso output será o conjunto total de dados paginado de 100 em 100 (Figura 18) artigos e outra query que irá retornar o artigo pelo id de forma a apresentar individualmente cada artigo e uma query que irá ler uma *string* e devolver todos os artigos que possuam essa mesma *string* (Figura 19).

```
// Returns Article list
module.exports.list = () => {
  return Article
    .find()
    .paginate(1,100)
    .sort({name:1})
    .exec()
}
```

Figura 18: Exemplo de uma Query em mongoDB que retorna todos os dados.

```
module.exports.lookupById = id => {
  return Article
    .findById(id)
    .exec()
}
```

Figura 19: Exemplo de uma Query em mongoDB que retorna um dado por id.

Apesar de estarmos a tratar do MVC, de forma a que efetivamente seja possível redirecionar múltiplas páginas na plataforma online foi necessário criar um script de *route* onde iríamos definir as rotas para cada página. Este *script* é um complemento ao Controller, visto que sem este passo não seria possível apresentar qualquer resultado no site.

Sendo assim na Figura 20 podemos contemplar as APIs efetivamente que serão realizadas à base de dados. Basicamente, neste processo está a acontecer um GET, que irá utilizar um controller específico e por sua vez irá devolver um conjunto de dados de acordo com o output esperado. Por exemplo no caso do segundo GET, irá devolver uma página com um artigo específico pelo seu id, sendo que o seu url é o id do artigo e, tudo isto utilizando o controller "lookupById" desenvolvido.

```

/* GET articles listing. */
router.get('/', (req,res)=>{
  Article.list()
    .then(data => res.render('articles', {articles: data}))
    .catch(err => res.render('error', {error: err}))
})

/* GET single article. */
router.get('/:id', (req, res) => {
  const { id } = req.params;
  Article.lookupById(id)
    .then(data => res.render('article', { article: data, id }))
    .catch(() => {
      req.flash('error', 'Cannot find that resource')
      res.redirect('/article')
    })
})

```

Figura 20: Script das routes com os GETs necessários

Uma vez concluído o Controller passamos para as Views (o que irá "renderizar" o Model em uma forma específica para a interação, geralmente uma interface de utilizador [9]) onde vamos, através do *pug.js*, criar o HTML das páginas necessárias. Desenvolvemos duas páginas, uma onde serão apresentados todos os artigos presentes no dataset (Figura 21), e uma segunda página onde será apresentado um artigo de forma detalhada (Figura 22).

```

block content
  div.w3-container.w3-teal.w3-margin-bottom.w3-margin-right.w3-margin-left.w3-margin-top
    h1.White Articles
    input(class="w3-input w3-margin-bottom", type="text", id="tagCode", onkeyup="filterTag()" placeholder="Search...")

  div.w3.w3-margin-right.w3-margin-left
    <table class="w3-table w3-striped w3-bordered w3-border">
      <thead class="w3-teal">
        <tr>
          <th style="width:50%"> Title </th>
          <th style="width:9%"> Publish Time </th>
          <th style="width:31%"> Authors </th>
          <th style="width:10%"> Journal </th>
        </thead>
        <tbody class="w3-white">
          <for article in articles>
            <tr>
              <td= article.title>
              <td= article.publish_time>
              <td= article.authors>
              <td= article.journal>
            </tr>
          </tbody>
        <a.btn.w3-btn(href="/article/" + article._id) Show Article>
      </table>

```

Figura 21: View de todos os artigos

```

block content
  div.w3-container.w3-teal.w3-margin-bottom.w3-margin-right.w3-margin-left.w3-margin-top
    h1.white= article.title

  div.w3.w3-margin-right.w3-margin-left
    <h2>Description</h2>
    <table class="w3-table w3-striped w3-bordered w3-border">
      <thead class="w3-teal"><th>Source</th><th>Doi</th><th>Pmcid</th><th>License</th><th>Publish Time</th></thead>
      tr
        td= article.source_x
        td= article.doi
        td= article.pmcid
        td= article.license
        td= article.publish_time
      </table>
      <table class="w3-table w3-striped w3-bordered w3-border">
      <thead class="w3-teal"><th>Authors</th><th>Journal</th></thead>
      tr
        td= article.authors
        td= article.journal
      </table>

    div.w3.w3-margin-right.w3-justify.w3-margin-left

      <h1>Abstract</h1>
      td= article.abstract
    </div>

  div.w3.w3-margin-right.w3-margin-left
    <h2>More Information</h2>
    <table class="w3-table w3-striped w3-bordered w3-border">
      <thead class="w3-teal"><th>cord_uid</th><th>sha</th><th>pubmed_id</th><th>mag_id</th><th>who_covidence_id</th></thead>
      tr
        td= article.cord_uid
        td= article.sha
        td= article.pubmed_id
        td= article.mag_id
        td= article.who_covidence_id
      </table>
      <table class="w3-table w3-striped w3-bordered w3-border">
      <thead class="w3-teal"><th>arxiv_id</th><th>pdf_json_files</th></thead>
      tr
        td= article.arxiv_id
        td= article.pdf_json_files
      </table>
      <table class="w3-table w3-striped w3-bordered w3-border">
      <thead class="w3-teal"><th>pmc_json_files</th><th>url</th><th>s2_id</th></thead>
      tr
        td= article.pmc_json_files
        td= article.url
        td= article.s2_id
      </table>

```

Figura 22: View de um artigo em específico

Para concluir esta Tarefa foi necessário desenvolver um sistema de pesquisa que permita pesquisar por palavras chave, autores, etc. Para tal foi desenvolvido um *script* (Figura 23) que irá ler a *string* introduzida pelo utilizador no *input* e irá devolver todos os casos em que possua a mesma string.

```

script.
let tagSearch = document.getElementsByClassName('ml')[0].children;
function filterTag() {
  let name = document.getElementById('tagCode').value.toUpperCase();
  let date = document.getElementById('tagCode').value.toUpperCase();
  let authors = document.getElementById('tagCode').value.toUpperCase();
  let journal = document.getElementById('tagCode').value.toUpperCase();
  for(let i = 0; i < tagSearch.length ; i+=1){
    //- console.log(tagSearch[i].children[0].innerText)
    if(tagSearch[i].children[0].innerText.toUpperCase().indexOf(name) > -1 ||
      tagSearch[i].children[1].innerText.toUpperCase().indexOf(date) > -1 ||
      tagSearch[i].children[2].innerText.toUpperCase().indexOf(authors) > -1 ||
      tagSearch[i].children[3].innerText.toUpperCase().indexOf(journal) > -1){
      tagSearch[i].style.display = ""
    }
    else tagSearch[i].style.display = "none"
  }
}
}

```

Figura 23: Script do sistema de pesquisa.

Existem várias formas de realizar um motor de busca sendo que uma delas seria ir diretamente à base de dados tal como as figuras 24 e 25 indicam. Tanto através de um *script*, como através de uma query é possível obter resultados muito semelhantes.

```

module.exports.lookupByHashtag = (title) => {
  return Article
    .find({ title: title })
    .exec()
}

```

Figura 24: Controller do sistema de pesquisa.

```

router.post('/title', (req, res) => {
  const { tag } = req.body
  if (tag == 'all') res.redirect('/articles')
  Article.lookupByHashtag(tag)
    .then(data => res.render('articles', { articles: data, title: tag }))
    .catch(err => res.render('error', { error: err }))
})

```

Figura 25: Route do sistema de pesquisa.

O resultado desta tarefa encontram-se nas Figuras seguintes (Figuras 26 a 31).

Articles					
Search...					
Title	Publish Time	Authors	Journal		
Studying copy number variations using a nanofluidic platform	2008-08-18	Qin, Jian; Jones, Robert C.; Ramakrishnan, Ramesh	Nucleic Acids Res	Show Article	
Immune reconstitution inflammatory syndrome (IRIS): review of common infectious manifestations and treatment options	2007-05-08	Murdoch, David M; Venter, Willem DF; Van Rie, Annelies; Feldman, Charles	AIDS Res Ther	Show Article	
Molecular evidence for the evolution of ichnoviruses from ascoviruses by symbiogenesis	2008-09-18	Bigot, Yves; Samain, Sylvie; Augé-Gouillou, Corinne; Federici, Brian A	BMC Evol Biol	Show Article	
Nitric oxide: a pro-inflammatory mediator in lung disease?	2000-08-15	Vliet, Albert van der; Eiserich, Jason P; Cross, Carroll E	Respir Res	Show Article	
Screening Pneumonia Patients for Mimivirus	2008-03-31	Dare, Ryan K.; Chittaganpitch, Malinee; Erdman, Dean D.	Emerg Infect Dis	Show Article	
Transmission Parameters of the 2001 Foot and Mouth Epidemic in Great Britain	2007-06-06	Chis Ster, Irina; Ferguson, Neil M.	PLoS One	Show Article	
WU Polyomavirus Infection in Children, Germany	2008-04-30	Neske, Florian; Blessing, Kerstin; Ullrich, Franziska; Pröttel, Anika; Kreth, Hans Wolfgang; Weissbrich, Benedikt	Emerg Infect Dis	Show Article	
Local public health workers' perceptions toward responding to an influenza pandemic	2006-04-18	Balicer, Ran D; Omer, Saad B; Barnett, Daniel J; Everly, George S	BMC Public Health	Show Article	

Figura 26: Página com todos os artigos.

Articles					
trans					
Title	Publish Time	Authors	Journal		
Transmission Parameters of the 2001 Foot and Mouth Epidemic in Great Britain	2007-06-06	Chis Ster, Irina; Ferguson, Neil M.	PLoS One	Show Article	
Debate: Transfusing to normal haemoglobin levels will not improve outcome	2001-03-08	Alvarez, Gonzalo; Hébert, Paul C; Szick, Sharyn	Crit Care	Show Article	
Factors affecting translation at the programmed -1 ribosomal frameshifting site of Cocksfoot mottle virus RNA in vivo	2005-04-20	Mäkeläinen, Katri; Mäkinen, Kristiina	Nucleic Acids Res	Show Article	

Figura 27: Resultado de um search por "trans".

Articles					
2001-07-04					
Title	Publish Time	Authors	Journal		
Clinical features of culture-proven Mycoplasma pneumoniae infections at King Abdulaziz University Hospital, Jeddah, Saudi Arabia	2001-07-04	Madani, Tariq A; Al-Ghamdi, Aisha A	BMC Infect Dis	Show Article	

Figura 28: Resultado de um search pela data "2001-07-04".

Articles				
Respir				
Title	Publish Time	Authors	Journal	
Nitric oxide: a pro-inflammatory mediator in lung disease?	2000-08-15	Vliet, Albert van der; Eiserich, Jason P; Cross, Carroll E	Respir Res	Show Article
Surfactant protein-D and pulmonary host defense	2000-08-25	Crouch, Erika C	Respir Res	Show Article
Surfactant therapy for acute respiratory failure in children: a systematic review and meta-analysis	2007-06-15	Duffett, Mark; Choong, Karen; Ng, Vivian; Randolph, Adrienne; Cook, Deborah J	Crit Care	Show Article

Figura 29: Resultado de um search por "Respir".

Articles				
vliet				
Title	Publish Time	Authors	Journal	
Nitric oxide: a pro-inflammatory mediator in lung disease?	2000-08-15	Vliet, Albert van der; Eiserich, Jason P; Cross, Carroll E	Respir Res	Show Article

Figura 30: Resultado de um search pelo autor "vliet".

Molecular evidence for the evolution of ichnoviruses from ascoviruses by symbiogenesis				
Description				
Source	Doi	Pmcid	License	Publish Time
PMC	10.1186/1471-2148-8-253	PMC2567993	cc-by	2008-09-18
Authors				Journal
Bigot, Yves; Samain, Sylvie; Augé-Gouillou, Corinne; Federici, Brian A				BMC Evol Biol
Abstract				
<p>BACKGROUND: Female endoparasitic ichneumonid wasps inject virus-like particles into their caterpillar hosts to suppress immunity. These particles are classified as ichnovirus virions and resemble ascovirus virions, which are also transmitted by parasitic wasps and attack caterpillars. Ascoviruses replicate DNA and produce virions. Polydnavirus DNA consists of wasp DNA replicated by the wasp from its genome, which also directs particle synthesis. Structural similarities between ascovirus and ichnovirus particles and the biology of their transmission suggest that ichnoviruses evolved from ascoviruses, although molecular evidence for this hypothesis is lacking. RESULTS: Here we show that a family of unique pox-D5 NTPase proteins in the Glypta fumiferanae ichnovirus are related to three Diadromus pulchellus ascovirus proteins encoded by ORFs 90, 91 and 93. A new alignment technique also shows that two proteins from a related ichnovirus are orthologs of other ascovirus virion proteins. CONCLUSION: Our results provide molecular evidence supporting the origin of ichnoviruses from ascoviruses by lateral transfer of ascoviral genes into ichneumonid wasp genomes, perhaps the first example of symbiogenesis between large DNA viruses and eukaryotic organisms. We also discuss the limits of this evidence through complementary studies, which revealed that passive lateral transfer of viral genes among polydnaviral, bacterial, and wasp genomes may have occurred repeatedly through an intimate coupling of both recombination and replication of viral genomes during evolution. The impact of passive lateral transfers on evolutionary relationships between polydnaviruses and viruses with large double-stranded genomes is considered in the context of the theory of symbiogenesis.</p>				
More Information				
cord_uid	sha	pubmed_id	mag_id	who_covidence_id
p56v8w1	9f9e925d9999ab39745f2ee8be3efffb5277d082	18801176		
arxiv_id	pdf_json_files			
	document_parses/pdf_json/9f9e925d9999ab39745f2ee8be3efffb5277d082.json			
pmc_json_files	url	s2_id		
document_parses/pmc_json/PMC2567993.xml.json	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2567993/			

Figura 31: Página individual com apenas 1 artigo.

5 Tarefa 3 - Text Mining

Nos dias de hoje, a forma como as pessoas se comunicam/partilham informação diz muito sobre as pessoas, tornando assim os conceitos de linguagem relevantes. No entanto, existem muitos idiomas no mundo e cada um tem os seus padrões e alfabetos. A combinação das diferentes palavras de uma forma organizada resulta na formação de frases. Cada idioma tem regras próprias para desenvolver frases, sendo estas também conhecidas como gramática [10].

Posto isto, e tendo em conta que a maioria dos dados gerados enquanto falamos, enviamos/escrevemos mensagens de texto (via WhatsApp, e-mail, Facebook, Instagram) são dados não estruturados é necessário alguma técnica que ajude a obter informação e ou conhecimento sobre estes dados não estruturados, ou seja, o *Text Mining* (em português Mineração de Texto). Desta feita, o *Text Mining* é o processo de derivar informações significativas a partir de texto, onde o principal objetivo é transformar os textos em dados para análise, por meio da aplicação do Processamento de Linguagem Natural (*Natural Language Processing - NLP*). O NLP é um componente do Text Mining que realiza um tipo especial de análise linguística que ajuda uma máquina a "ler" texto. Ele usa uma metodologia diferente para decifrar as ambiguidades na linguagem humana [11, 12].

Desta forma, e tendo em conta a natureza dos nossos dados, a terceira, e última tarefa, consiste em desenvolver um modelo de *Machine Learning* que permita fazer *Text Mining* aos *abstracts*.

Antes de desenvolvermos o modelo de machine, tivemos que fazer uma preparação dos dados comum a todos os algoritmos de *Text Mining*. Começamos por selecionar apenas as colunas que nos interessavam, ficando apenas *title*, *abstract*, *publish_time*, *authors* e *journal* (Figura 32). As restantes colunas foram eliminadas uma vez que (como já vimos na descrição e análise dos dados) existem muitos valores em falta (*missing values*). De seguida, eliminamos os *missing values* das colunas selecionadas. Optamos por eliminar, uma vez que se trata de artigos científicos não fazia sentido recuperar/substituir os dados em falta (Figura 33).



```
#Remove some columns
papers = papers.drop(columns=['mag_id', 'who_covidence_id', 'arxiv_id', 'pdf_json_files', 'pmc_json_files', 'url', 's2_id', 'sha', 'cord_uid', 'source_x', 'doi', 'pmcid', 'pubmed_id', 'license'])
papers.head()
```

	title	abstract	publish_time	authors	journal
19247	Scientific Abstracts	NaN	2011-12-30	NaN	Reprod Sci
245998	Troubleshoot It: Accuracy of Various Thermomet...	NaN	2020	Crossley, Becky	Biomed Instrum Technol
529545	DetectaWeb-Distress Scale: A Global and Multid... Emotional disorder symptoms are highly prevale...	2021-02-15	Piqueras, Jose A.; Garcia-Olcina, Mariola; Riv...		Front Psychol
461936	Importance of inclusion of pregnant and breast... Investigators are employing unprecedented inno...	2020-04-15	LaCourse, Sylvia M; John-Stewart, Grace; Adams...		Clin Infect Dis
131191	COVID-19 and Long-Term Care Policy for Older P... Hong Kong is a major international travel hub ...	2020-05-31	Lum, Terry; Shi, Cheng; Wong, Gloria; Wong, Kayla		Journal of aging & social policy

Figura 32: Remover algumas colunas.

```
#remove missing values
papers = papers.dropna()
papers.head()
```

	title	abstract	publish_time	authors	journal
529545	DetectaWeb-Distress Scale: A Global and Multid...	Emotional disorder symptoms are highly prevale...	2021-02-15	Piqueras, Jose A.; Garcia-Olcina, Mariola; Riv...	Front Psychol
461936	Importance of inclusion of pregnant and breast...	Investigators are employing unprecedented inno...	2020-04-15	LaCourse, Sylvia M.; John-Stewart, Grace; Adams...	Clin Infect Dis
131191	COVID-19 and Long-Term Care Policy for Older P...	Hong Kong is a major international travel hub ...	2020-05-31	Lum, Terry; Shi, Cheng; Wong, Gloria; Wong, Kayla	Journal of aging & social policy
569656	Public perceptions of non-pharmaceutical inter...	BACKGROUND: Non-pharmaceutical public health ...	2014-06-11	Teasdale, Emma; Santer, Miriam; Geraghty, Adam...	BMC Public Health
337995	Telling the Truth to Child Cancer Patients in ...	A notable feature of the COVID-19 pandemic is ...	2020	Gillam, Lynn; Spriggs, Merle; Delany, Clare; C...	J Bioeth Inq

Figura 33: Remover valores em falta.

Posteriormente, removemos a pontuação (*punctuation*) e transformamos o texto da coluna *abstract* em minúsculas (*lower*). Isto é importante, uma vez que torna os dados mais acessíveis para uma análise e resultados confiáveis (Figura 34).

```
#Load the regular expression library
import re

#Remove punctuation
papers['abstract'].map(lambda x: re.sub('[\.,!?:()-;:]', '', x))
papers['authors'].map(lambda x: re.sub('[,;.-]', '', x))
papers['title'].map(lambda x: re.sub('[,;).-:]', '', x))
papers['journal'].map(lambda x: re.sub('[,;).-:]', '', x))

#Convert the abstract to lowercase
papers['abstract'] = papers['abstract'].str.lower()
#papers['abstract'].map(lambda x: x.lower())

papers['abstract'].head()

529545    emotional disorder symptoms are highly prevale...
461936    investigators are employing unprecedented inno...
131191    hong kong is a major international travel hub ...
569656    background: non-pharmaceutical public health i...
337995    a notable feature of the covid-19 pandemic is ...
Name: abstract, dtype: object

papers.head()
```

Figura 34: Remover pontuação.

Seguidamente, decidimos fazer uma nuvem de palavras usando o pacote *wordcloud* para obter uma representação visual das palavras mais comuns. Esta nuvem de palavras foi importante para entender os dados e garantir que estamos no caminho certo e se mais algum pré-processamento seria necessário antes de treinar o modelo (Figura 35).

Depois de terminar a preparação dos dados, desenvolvemos um modelo, nomeadamente o modelo LDA *Latent dirichlet allocation*. O LDA é um modelo probabilístico generativo que assume que cada tópico é uma mistura de um conjunto de palavras subjacentes e que cada documento é uma mistura de um conjunto de probabilidades de tópicos (Figura 37). Podemos descrever o processo de LDA como, dado o número M de documentos, o número N de palavras e número K anterior de tópicos, o modelo treina para produzir:

- ψ , a distribuição de palavras para cada tópico K ;
- ϕ , a distribuição de tópicos para cada documento i .

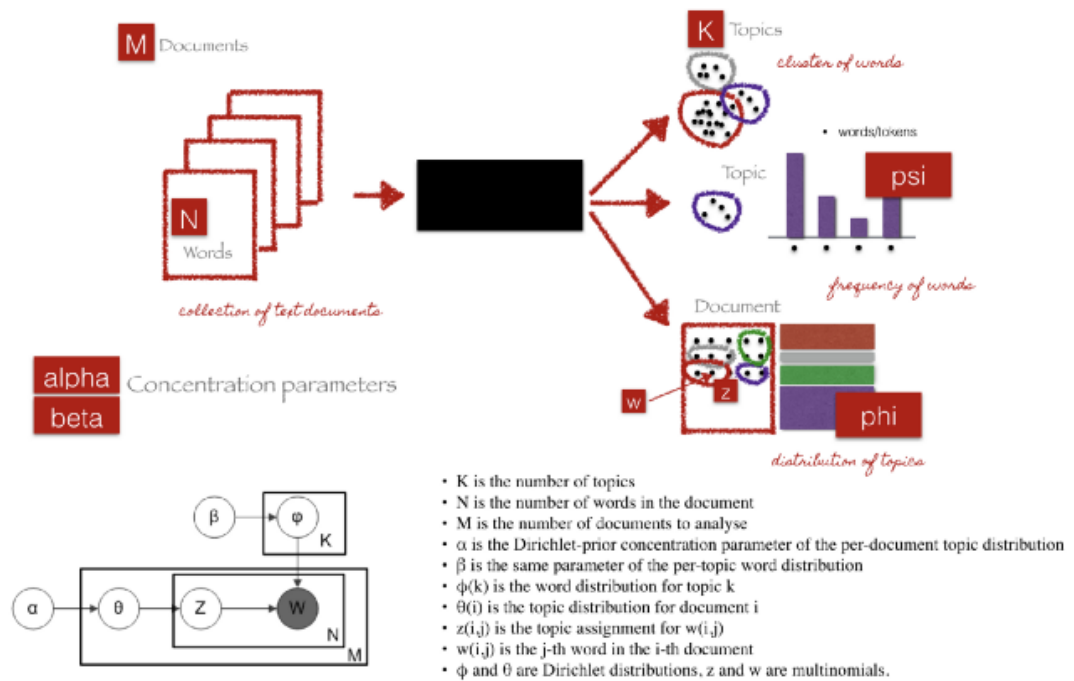


Figura 37: Representação do Modelo LDA.

O parâmetro alfa é o parâmetro de concentração anterior de *Dirichlet* que representa a densidade de tópico do documento - com um alfa mais alto, os documentos são compostos por mais tópicos e resultam numa distribuição de tópicos mais específica por documento.

O parâmetro beta é o mesmo parâmetro de concentração anterior que representa a densidade de palavras do tópico - com beta alto, os tópicos são considerados compostos pela maioria das palavras e resultam numa distribuição de palavras mais específica por tópico.

Para o desenvolvimento do nosso modelo LDA, e de forma a manter as coisas simples, optamos por manter os parâmetros padrão, excepto o número de tópicos. No

nosso caso, construímos um modelo com 10 tópicos onde cada tópico é uma combinação de palavras-chave, e cada palavra-chave contribui com um determinado peso para o tópico (Figura 38).

```
from pprint import pprint

#number of topics
num_topics = 10

#Build LDA model
lda_model = gensim.models.LdaMulticore(corpus=corpus, id2word=dicword, num_topics=num_topics)

#Print the Keyword in the 10 topics
pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]
```

```
[(0,
  '0.009*"patients" + 0.008*"covid" + 0.005*"study" + 0.005*"disease" + '
  '0.005*"psychiatrists" + 0.005*"technology" + 0.004*"impact" + '
  '0.004*"analysis" + 0.004*"potential" + 0.004*"studies"'),
 (1,
  '0.010*"covid" + 0.006*"care" + 0.005*"children" + 0.005*"risk" + '
  '0.004*"health" + 0.003*"stenosis" + 0.003*"effects" + 0.003*"associated" + '
  '0.003*"may" + 0.003*"vein"'),
 (2,
  '0.011*"covid" + 0.010*"cov" + 0.009*"sars" + 0.008*"disease" + 0.008*"rna" '
  '+ 0.006*"wastewater" + 0.005*"viral" + 0.005*"infection" + 0.005*"based" + '
  '0.004*"study"'),
 (3,
  '0.008*"covid" + 0.007*"pets" + 0.007*"viral" + 0.006*"infection" + '
  '0.005*"host" + 0.005*"related" + 0.004*"may" + 0.004*"iav" + 0.004*"higher" '
  '+ 0.004*"pandemic"'),
 ...]
```

Figura 38: Modelo LDA desenvolvido.

Depois do modelo treinado, visualizamos os tópicos para interpretabilidade. Para fazer isso, utilizamos o pacote de visualização *pyLDavis*, uma vez que este foi projetado, de forma iterativa, com:

1. Melhor compreensão e interpretação dos tópicos individuais;
2. Compreender melhor a relação entre os diferentes tópicos.

Para o ponto 1, podemos selecionar manualmente cada tópico e visualizar os principais termos mais frequentes e/ou relevantes variando os valores do parâmetro λ . Este processo é importante quando estamos a tentar atribuir um nome interpretável ou significado humano para cada tópico.

Para o ponto 2, podemos explorar o gráfico de forma iterativa e ajuda-nos a aprender sobre como os tópicos se relacionam entre si, e inclui uma possível estrutura de nível superior entre os tópicos.

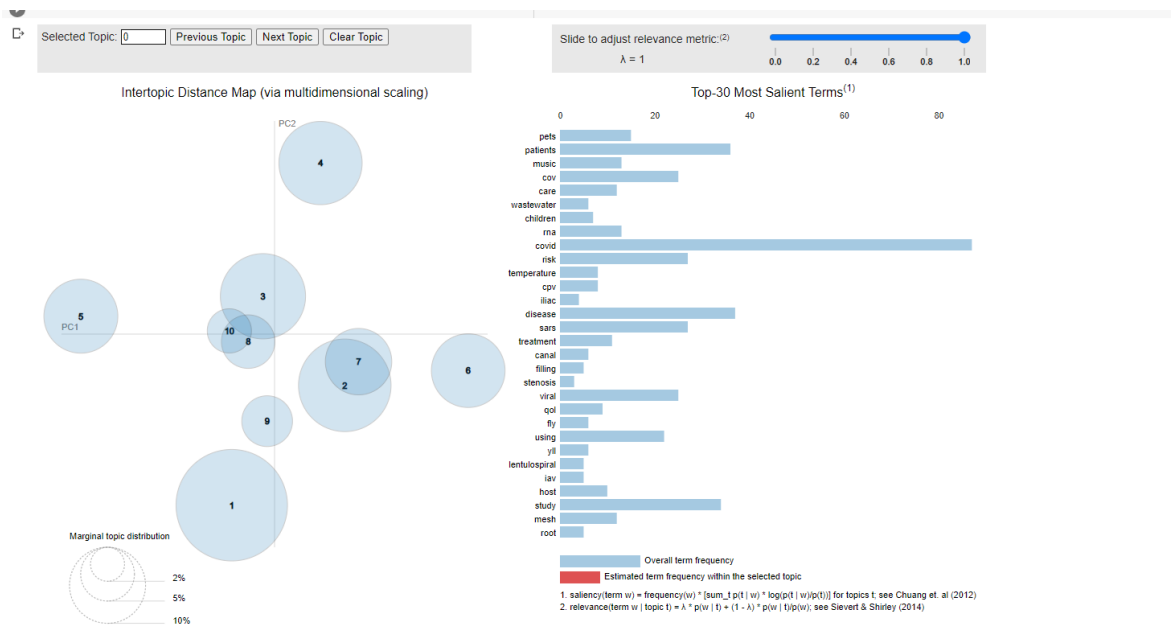


Figura 39: Visualização dos tópicos para interpretação.

Ao interpretar os resultados obtidos do modelo LDA concluímos que:

- a palavra covid é quase sempre a palavra mais representativa nos diferentes tópicos para $\lambda = 1$;
- ao alterarmos os valores do λ temos resultados bastante diferentes;
- os tópicos com mais tokens são: tópico 1 (21.5%), tópico 2 (14.8%) e tópico 3 (12.6%);
- existem alguns tópicos que se relacionam entre si, nomeadamente os tópicos 3, 10 e 5 e ainda os tópicos 2 e 7;
- em alguns tópicos temos algumas palavras que podem representar o mesmo (por exemplo no tópico temos covid, cov, sars e disease que aparentemente devem fazer referência à mesma doença no caso covid-19);
- nos tópicos 8, 9 e 10 a estimativa da frequência das palavras é praticamente constante, ou seja, para este tópicos estas palavras aparecem quase sempre com a mesma frequência.

Para uma melhor análise aconselhamos a iteragir com o programa desonvolvido.

Para além do modelo LDA, e uma vez que achamos importante, desenvolvemos uma técnica em que dada um conjunto de questões (previamente feitas) é capaz de devolver quais os artigos que melhor respondem a cada uma das questões. Para responder as

estas questões, definimos um conjunto de palavras-chave para cada questão e pesquisamos pelas palavras-chave. Para além disso, e tendo em conta os objetivos do conjunto de dados (definido na descrição e análise de dados) selecionamos apenas os artigos que tinham as palavras *covid*, *-cov-2*, *cov2* e *ncov* (Figura 40 e Figura 41).

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# keep only documents with covid -cov-2 and cov2
def search_focus(df):
    dfa = df[df['abstract'].str.contains('covid')]
    dfb = df[df['abstract'].str.contains('-cov-2')]
    dfc = df[df['abstract'].str.contains('cov2')]
    dfd = df[df['abstract'].str.contains('ncov')]
    frames=[dfa,dfb,dfc,dfd]
    df = pd.concat(frames)
    df=df.drop_duplicates(subset='title', keep="first")
    return df

# load the meta data from the CSV file using some columns (abstract, title, authors),
df=pd.read_csv('metadata.csv', usecols=['title','journal','abstract','authors','doi','publish_time','sha'],engine='python',encoding='utf-8', error_bad_lines=False)

#drop duplicates
#df=df.drop_duplicates()
#drop NANS
df=df.fillna('no data provided')
df = df.drop_duplicates(subset='title', keep="first")
df=df[df['publish_time'].str.contains('2020')]
# convert abstracts to lowercase
df['abstract'] = df['abstract'].str.lower()+df['title'].str.lower()
#show 5 lines of the new dataframe
df=search_focus(df)
print (df.shape)
df.head()
```

Figura 40: Selecionar os artigos com as palavras *covid*, *-cov-2*, *cov2* e *ncov*.

1: What is the range of incubation periods for the disease in humans?			
pub_date	authors	title	excerpt
2020-03-08	Yang, et al.	[The preliminary analysis on the characteristics of the cluster for the Corona Virus Disease].	We selected 325 cases to estimate the incubation period and found its range is 1 to 20 days, median was 7 days, and mode was 4 days.
2020-12-01	Chen, et al.	Epidemiological analysis of 18 patients with COVID-19.	The epidemiological characteristics were as follows: (1) the median incubation period was 8 days (with an interquartile range of 4-12 days); (2) the incubation period in one case was ≥18 days; (3) one infant patient was asymptomatic prior to their diagnosis; and (4) two asymptomatic patients had a positive nucleic acid test after their family members were diagnosed with covid-19.
2: How long are individuals are contagious?			
pub_date	authors	title	excerpt
2020-12-01	Pan, et al.	Retrospective analysis of the effect of current clinical medications and clinicopathological factors on viral shedding in COVID-19 patients.	Detailed clinical data of each patient were collected, and the factors that affected the duration of viral shedding were retrospectively analysed. The median duration of viral shedding in the 186 covid-19 patients was 13 days. The median duration of viral shedding was 12 days in non-severe patients, and 17 days in severe patients, and there was a significant difference between the two groups (p<0.001). Spearman's rank correlation analysis showed that the onset-drug interval was positively correlated with the duration of viral shedding (r=0.446; p<0.0001). Lpvir shortened the duration of viral shedding, and the smaller the interval between presentation and lpv/r onset was, the faster viral shedding occurred retrospective analysis of the effect of current clinical medications and clinicopathological factors on viral shedding in covid-19 patients.
2020-07-21	Dodds, et al.	Model-Informed Drug Repurposing: Viral Kinetic Modeling to Prioritize Rational Drug Combinations for COVID-19.	The endpoints and metrics included viral load area under the curve (auc), duration of viral shedding, and epithelial cells infected. In addition, we observed that the time-window opportunity for a therapeutic intervention to effect duration of viral shedding exceeds the effect on sparing epithelial cells from infection or impact on viral load auc.

Figura 41: Resultados obtidos para algumas questões.

6 Conclusão

Este relatório aborda algumas partes fundamentais de tudo o que foi o desenvolvimento ao longo do projeto. Neste trabalho pudemos adquirir mais conhecimento, competências, capacidades e experiência tanto nas áreas de Desenvolvimento Web como de Machine Learning visto estarmos perante um projeto desafiante do qual nunca tínhamos trabalhado antes.

Com este trabalho percebemos a importância das plataformas online e de que forma estas nos ajudam a ser mais eficientes e a produzir resultados com maior qualidade. Para além disso, compreendemos de que forma o processamento de linguagem natural pode ser útil para o dia a dia, nomeadamente para a descoberta de novos *insights* sobre a pandemia que nos afeta. Através do modelo LDA conseguimos descobrir quais os termos mais frequentes nos diferentes tópicos e o relacionamento entre os diferentes tópicos. Com os resultados obtidos, e dada a natureza obtida acreditamos que este modelo teve um bom desempenho.

6.1 Trabalho Futuro

Como trabalho futuro podíamos melhorar a plataforma online criando uma visualização mais *User Friendly* para o utilizador e adicionando mais métodos de pesquisa e visualização dos dados. Desenvolver e implementar um *front-end* para o Text Mining em que o utilizador seria capaz de inserir os dados num formulário e receber os resultados do modelo. Implementar uma página Perguntas mais Frequentes, e com base na técnica desenvolvida apresentar os resultados para cada pergunta.

Por último, desenvolver outros modelos de *Text Mining* e comparar os resultados com o modelo LDA desenvolvido. Eventualmente tentar fazer *Text Mining Classification*, e/ou até mesmo desenvolver um sistema de recomendação.

Referências

- [1] C. Faloutsos, “Foreword,” in *Data Mining (Third Edition)* (J. Han, M. Kamber, and J. Pei, eds.), The Morgan Kaufmann Series in Data Management Systems, pp. xix–xx, Boston: Morgan Kaufmann, third edition ed., 2012.
- [2] V. Kotu and B. Deshpande, “Chapter 8 - model evaluation,” in *Predictive Analytics and Data Mining* (V. Kotu and B. Deshpande, eds.), pp. 257–273, Boston: Morgan Kaufmann, 2015.
- [3] M. Azure, “Banco de dados nosql – o que é nosql?.” <https://azure.microsoft.com/pt-br/overview/nosql-database/>. Acedido em: 22-06-2021.
- [4] MongoDB, “The database for modern applications.” <https://www.mongodb.com/>. Acedido em: 22-06-2021.
- [5] MongoDB, “mongod.” <https://docs.mongodb.com/manual/reference/program/mongod/>. Acedido em: 22-06-2021.
- [6] Node.js, “Node.” <https://nodejs.org/en/>. Acedido em: 22-06-2021.
- [7] Express, “Express fast, unopinionated, minimalist web framework for node.js.” <https://expressjs.com/>. Acedido em: 22-06-2021.
- [8] Pug.js, “What is pug.js (jade) and how can we use it within a node.js web application?.” <https://www.codeburst.io/what-is-pug-js-jade-and-how-can-we-use-it-within-a-node-js-web-application-69a>. Acedido em: 22-06-2021.
- [9] J. Lamim, “Mvc - o padrão de arquitetura de software.” https://www.oficinadanet.com.br/artigo/1687/mvc_-_o_padrao_de_arquitetura_de_software. Acedido em: 22-06-2021.
- [10] S. Kapadia, “Topic modeling in python: Latent dirichlet allocation (lda).” <https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3>. Acedido em: 26-06-2021.
- [11] A. Navlani, “Text analytics for beginners using nltk.” <https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk>. Acedido em: 26-06-2021.
- [12] D. Subramanian, “Text mining in python: Steps and examples.” <https://towardsai.net/p/data-mining/text-mining-in-python-steps-and-examples-78b3f8fd913b>. Acedido em: 26-06-2021.