

UNIVERSIDADE DO MINHO

MESTRADO EM ENGENHARIA INFORMÁTICA
SPLN

Ferramenta para Anonimização de Dados



Henrique Parola
Franisco pg50415



José Pedro
pg50525



pgTODO



Conteúdo

1	Introdução	2
1.1	Contexto	2
1.2	Propósito e Objetivos	2
2	Sistema	3
2.1	Arquitetura	3
2.2	Anonimização dos Nomes	3
2.3	Anonimização dos Endereços	6
3	Modos de Uso	10
4	Conclusão	11



1 Introdução

1.1 Contexto

A privacidade de um indivíduo está intimamente ligada aos seus dados. Dados os quais, nos dias de hoje, são gerados numa quantidade nunca antes vista através da Internet. Estas dados, quando não são cuidadosamente tratados, podem afetar a segurança das pessoas. Neste segmento que surge o conceito de anonimização dos dados.

Uma definição conceptual da anonimização de dados pode ser “para anonimizar quaisquer dados, têm de lhes ser retirados elementos suficientes para que deixe de ser possível identificar (de forma irreversível) o titular dos dados” [1]. Neste contexto surge o Regulamento Geral de Proteção de Dados (RGPD), como também a A Lei Geral de Proteção de Dados Pessoais (LGPD) que define um dado anonimizado aquele que, originalmente, era relativo a uma pessoa, mas que passou por etapas que garantiram a desvinculação dele a essa pessoa [2]. Conforme a LGPD, alguns exemplos de dados pessoais são: nome, CPF, e-mail, idade, profissão, foto, entre outros [3].

1.2 Propósito e Objetivos

O propósito deste projeto é garantir a anonimização dos dados sensíveis presentes em documentos. Assim, o seu objetivo é concretizar um *software* que realiza uma série de tratamento de dados por forma a desvincular os dados das pessoas identificadas por eles. Este tratamento de dados resultará numa conversão de um dado documento numa versão sua anonimizada. Dentre as diversas formas de dados pessoais existentes, foram consideradas: o nome das pessoas/organizações, endereços (físicos ou na Web) e números identificadores de documentos (como o CC, carta de condução, entre outros).

Em suma, o objetivo principal do sistema é garantir a segurança dos indivíduos através de processos de anonimização. Uma vez que existem diversas formas de se concretizar esta tarefa, nas palavras de [1]: *não havendo um processo único de anonimização, a solução ideal será a que apresente em cada processo a maior impossibilidade da “re-identificação dos titulares dos dados”. Por princípio, a anonimização deverá ser um processo irreversível, análogo à destruição.*

2 Sistema

2.1 Arquitetura

O HShield é um programa em Python dividido em três módulos. Cada módulo é responsável por anonimizar um aspecto do texto de *input*. Estes aspectos foram divididos em **nomes**, endereços e **documentos**.

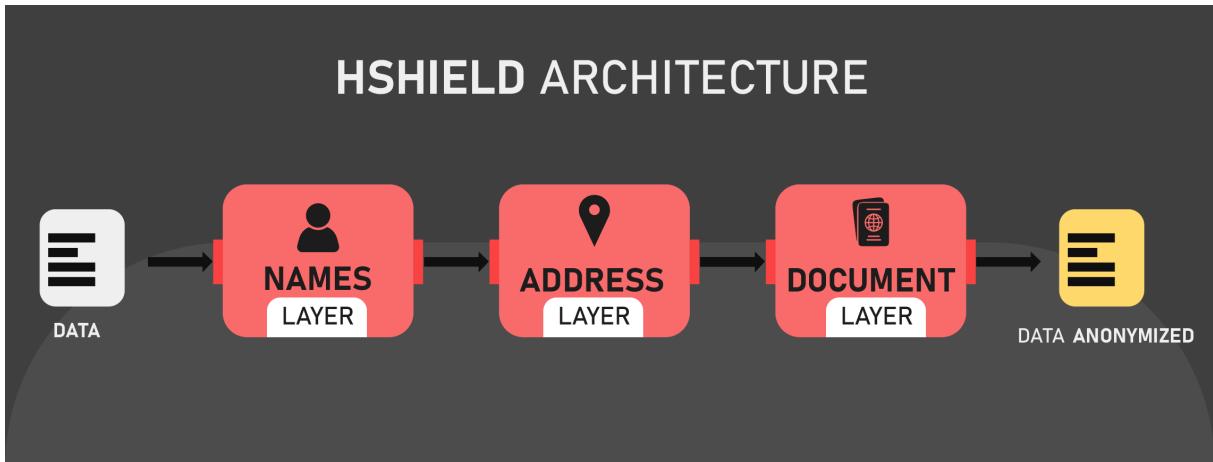


Figura 1: HShield - Arquitetura

2.2 Anonimização dos Nomes

Os ““nomes” anonimizados nesta etapa da ferramenta consistem em:

- **Nome de Pessoas:** nomes próprios, apelidos, alcunhas. Por exemplo: *John Smith, Sarah.*
- **Nomes de Organizações:** nomes de entidades estruturadas, geralmente composta por um grupo de pessoas, que trabalha para atingir objetivos específicos. Por exemplo: *Acme Industries, World Wildlife Fund.*

Para a anonimização destes nomes foi decidido que, ao serem identificados, deviam ser substituídos pelas correspondentes iniciais intercaladas com o "ponto final". Mas não somente isso, uma vez que poderia haver a problemática de nomes diferentes correspondem às mesmas iniciais. Ora, uma coisa é anonimizar um texto, outra é fazer ele perder o sentido e coesão. Não pretendemos causar a perda de significado no texto, isto é, deve ser possível continuar a lê-lo sem deixar de compreendê-lo. O seguinte texto é um exemplo disto:

José Pedro esteve na Praça dos Arsenalistas naquela tarde. Quando José Pedro encontrou João Pinto, já era tarde demais. João Pinto estava morto diante de José Pedro. A partir deste dia a vida de José Pedro nunca foi a mesma, nem Joana Pedrosa (sua parceira de trabalho no Banco do Brasil) acreditava mais nele.



Reparemos que as três entidades neste texto possuem as mesmas iniciais. Se os seus nomes apenas fossem substituídos pelas respectivas iniciais intercaladas com ponto, então teríamos uma frase do tipo:

Quando J.P encontrou J.P, já era tarde demais.

A compreensão da frase foi comprometida e, por tanto, um cuidado adicional deve ser tomado. Esse cuidado consiste justamente em adicionar um identificador numérico as entidades, por forma a, quando for decteado o José Pedro, seja possível distingui-lo do João Pinto. Este identificador, ao ser adicionado na anonimização, torna o texto do seguinte modo:

J.P(0) esteve na Praça dos Arsenalistas naquela tarde. Quando J.P(0) encontrou J.P(1), já era tarde demais. J.P(1) estava morto diante de J.P(0). A partir deste dia a vida de J.P(0) nunca foi a mesma, nem J.P(2) (sua parceira de trabalho no B.d.B(0)) acreditava mais nele.

Como pode ser visto, sem revelar a identidade de nenhuma entidade, consegue-se agora compreender a natureza do significado das frases.

A metodologia do algoritmo geral para anonimização dos nomes baseou-se então em 4 etapas:

1. Deteção das entidades do texto.
2. Filtragem das entidades que representam **pessoas e organizações**;
3. Calculo do identificador das entidades filtradas;
4. Substituição do nome destas entidades pelo seu nome anonimizado.

A etapa 1 e 2 foram concretizadas utilizando a biblioteca Spacy. Após carregar o modelo de processamento de texto no idioma do texto dado pelo utilizador e aplicar os processamentos linguísticos deste modelo no texto, é possível extrair as suas entidades da seguinte maneira:

```
1 nlp = spacy.load('en_core_web_sm')
2 doc = nlp(self.text)
3 for ent in doc.ents:
4     if ent.label_ == "PERSON" or ent.label_ == "ORG":
5         # substitution
```

Etapas 1 e 2 da Anonimização dos Nomes

Por outro lado, a etapa 3 e 4 foram realizadas com técnicas de processamento de texto em Python, utilizando alguns métodos sobre *strings* e o módulo RE para expressões regulares. Dada uma entidade, os seus respectivos nomes foram separados a partir de um ou mais carácter de espaço em branco e, de seguida, o nome anonimizado foi formado com as iniciais de cada nome concatenadas com o carácter "..". Inicialmente, foi utilizada a função split sobre strings com um único delimitador de "espaço" para serem separados



os nomes de uma entidade. Porém, esta alternativa não foi seguida, uma vez que alguns textos mal formados poderiam conter mais de um carácter de espaço em branco entre os nomes de uma entidade. Por isso que, como pode ser visto no excerto de código seguinte, foi utilizado o *split* do módulo RE que possibilitar a utilização de uma expressão regular para informar o delimitador do *split*.

```
1 if ent.label_ == "PERSON" or ent.label_ == "ORG":  
2     ent_names = re.split(r"\s+",ent.text)  
3     anonymized_name = ".".join(name[0] for name in ent_names)
```

Etapa 3 e 4 da Anonimização dos Nomes (parte 1)

Como foi exemplificado anteriormente, este termo `anonymized_name` ainda não está completo. Falta a adição do identificador da entidade. Para isto foi criado um dicionário chamado `dic_names`. Seu objetivo é relacionar, para cada valor de letras iniciais intercaladas com ponto (`anonymized_name` sem identificador) os nomes das entidades que utilizam tais letras iniciais de forma idêntica. Por outras palavras, no caso do exemplo anterior, este dicionário teria:

```
dic_names = {  
    "J.P" : ["José Pedro", "João Pinto", "Joana Pedrosa"]  
}
```

Etapa 3 e 4 da Anonimização dos Nomes (parte 2)

Deste modo, foi decidido que o identificador de cada entidade seria justamente a sua posição na lista de nomes do seu nome anonimizado. Daí resulta no José Pedro ser o J.P(0), o João Pinto ser o J.P(1) e a Joana Pedrosa a J.P(2). O algoritmo que efetua o cálculo do identificador é o seguinte:

```
1 if anonymized_name in dic_names:  
2     if ent.text in dic_names[anonymized_name]:  
3         id = dic_names[anonymized_name].index(ent.text)  
4     else:  
5         id = len(dic_names[anonymized_name])  
6         dic_names[anonymized_name].append(ent.text)  
7 else:  
8     id = 0  
9     dic_names[anonymized_name] = [ent.text]
```

Etapa 3 e 4 da Anonimização dos Nomes (parte 3)

Uma vez em posse do nome anonimizado com o seu identificador, bastava substituir todas as ocorrências do nome sem anonimização pelo termo anonimizado.

```
anonymized_name += ' ('+str(id)+')'  
self.text = re.sub(ent.text,anonymized_name,self.text)
```

Etapa 3 e 4 da Anonimização dos Nomes (parte 4)



2.3 Anonimização dos Endereços

Já nesta etapa, o objetivo passou para anonimizar endereços, quer endereços *web*, quer endereços físicos. Assim, os endereços anonimizados consistem em:

- **Endereços de Localização:** endereços físicos de locais no mundo. Por exemplo: *Rua Chãozinha, nº23*;
- **Endereços de email:** endereços de correio eletrónico. Por exemplo: *email@example.com*;
- **Endereços URL:** estes encontram-se subdivididos em duas partes, que serão tratadas de forma diferente:
 - **Endereços de Redes Sociais:** endereços de aplicações *web* muito conhecidas. Por exemplo: *www.facebook.com*;
 - **Endereços Web:** todos os endereços URL que não são de uma rede social conhecida. Por exemplo: *pt.overleaf.com*.

De forma a anonimizar os endereços, foi seguida a política definida no trabalho prático. Desta forma, quando um endereço de localização é encontrado, o mesmo é substituído por *localização...*, já quando um endereço de *email* é encontrado, este é substituído por *email...*, por outro lado, quando um endereço de rede social é encontrado, este é substituído pelo nome da rede social em questão seguido de reticências, por exemplo, *Instagram...*, por fim, quando um endereço *web* é encontrado, este é substituído por *www....*

Vejamos o seguinte exemplo:

Era uma bela manhã de verão, quando o José Pedro decidiu que iria visitar a Rua da Chãozinha, nº25, 1º andar, em Lisboa. Isto deveu-se ao anúncio que ele encontrou em www.instagram.com. Inicialmente, o José Pedro ainda visitou o vídeo presente em www.youtube.com para verificar a veracidade dos factos apresentados no anúncio. Como parecia tudo muito bom, dirigiu-se a www.google.com, para aceder ao seu email. Lá, enviou um email para reservas@gmail.com para reservar o seu lugar.

Ao aplicarmos a anonimização definida, obtemos o seguinte resultado:

Era uma bela manhã de verão, quando o José Pedro decidiu que iria visitar a localização... . Isto deveu-se ao anúncio que ele encontrou em www.... Inicialmente, o José Pedro ainda visitou o vídeo presente em www... para verificar a veracidade dos factos apresentados no anúncio. Como parecia tudo muito bom, dirigiu-se a www..., para aceder ao seu email. Lá, enviou um email para email... para reservar o seu lugar.

A metodologia do algoritmo geral para anonimização baseou-se então nas seguintes etapas:

1. Detecção dos endereços no texto;
2. Filtragem dos tipos de endereço;



3. Substituição dos *tokens* pelo seu valor anonimizado.

À semelhança do módulo anterior, foi utilizada a biblioteca Spacy para fornecer alguma ajuda na concretização dos objetivos propostos. O carregamento do modelo de processamento de texto para posteriormente aplicar os processamentos linguísticos do modelo no texto, sendo possível efetuar o tratamento pretendido é feito da seguinte maneira:

```
replace_loc = False
prev_token_space = False
for (i, token) in enumerate(doc):
    if token.like_email:
        # trata email
    elif token.like_url:
        # trata urls web e de redes sociais
    elif token.ent_type_ == "LOC" or token.ent_type_ == "GPE":
        # trata endereços de localização
```

Algoritmo geral

Por outro lado, foi ainda necessário a utilização do módulo RE para o tratamento da diferenciação entre urls genéricos e urls de redes sociais, bem como para a aglomeração de elementos pertencentes a uma localização (por exemplo, *Rua da Veiga, nº23, 5230-021* deverá ser substituído por um único parâmetro *localização...*).

O tratamento da diferenciação entre urls é realizado da seguinte maneira:

```
if token.like_url:
    url_matched = False
    for pattern, replacement in self.social_networks_regex.items():
        if re.search(pattern, token.text, re.IGNORECASE):
            url_matched = True
            # trata endereço de rede social
    if not url_matched:
        # trata endereço geral
```

Diferenciação de URLs

Assim, de forma a ser possível testar as expressões regulares das diferentes redes sociais detetadas, foi implementado um dicionário que associa a cada expressão regular o valor que o token deverá tomar caso dê *match* com a mesma:

```
social_networks_regex = {
    r"https://(?:www\.)?github\.com/([^\?#]+)": "GitHub...",
    r"https://(?:www\.)?gitlab\.com/([^\?#]+)": "GitLab...",
    r"""\https://(?:www\.)?goodreads\.com
        /(?:book/show/author/show/user/show)/(\d+)"": "
    "Goodreads...",  
...  
}
```

Dicionário com expressões regulares para redes sociais

As redes consideradas para substituições específicas foram:



- Facebook;
- Twitter;
- Instagram;
- LinkedIn;
- YouTube;
- Telegram;
- WhatsApp;
- TikTok;
- Pinterest;
- Reddit;
- Tumblr;
- Flickr;
- Quora;
- Medium;
- Twitch;
- Zoom;
- Google Meet;
- Jitsi;
- Trello;
- Slack;
- Discord;
- Stack Exchange;
- Stack Overflow;
- Stack Apps;
- GitHub;
- GitLab;
- Goodreads.



Por fim, de forma a efetuarmos o tratamento adequado da localização, é preciso analisar o contexto envolvente às palavras detetadas como localização, desta forma, o tratamento é efetuado da seguinte maneira:

```
if replace_loc:
    if (re.match(r"(\d+|em|na|no)", token.text)):
        if (self.check_context(doc, i)):
            continue
        elif (
            token.ent_type_ == "LOC"
            or token.ent_type_ == "GPE"
            or self.match_address(token.text)
        ):
            continue
        else:
            replace_loc = False
...
elif token.ent_type_ == "LOC" or token.ent_type_ == "GPE":
    if not replace_loc:
        replace_loc = True
    anonymized_text += "localização..."
```

Tratamento da localização

O método que permite a verificação de contexto é o seguinte:

```
def check_context(self, doc: spacy._doc_, i: int) -> bool:
    if i == 0:
        return False
    if self.match_address(doc[i-1].text) or
        self.match_address(doc[i+1].text):
        return True
    if doc[i + 1].ent_type_ == "LOC" or doc[i + 1].ent_type_ == "GPE":
        return True

def match_address(self, text: str) -> bool:
    for item in self.address_regex:
        if re.search(item, text, re.IGNORECASE):
            return True
    return False
```

Verificação de contexto para localização

Para isto, à semelhança daquilo que foi feito com o caso das redes sociais, possuímos uma lista com expressões regulares indicadoras de endereço que o Spacy não é capaz de detetar, visto dependerem do contexto envolvente:

```
address_regex = [
    r"n(([u|ú]m)?e(ro)?)?o?\.?\s?\d+",
    r"\d{4}-\d{2,3}-?",
    r"[,;:-]"
]
```

Dicionário para contexto de localização



3 Modos de Uso

A utilização do programa HShield pode englobar tanto uma anonimização global do documento de *input*, tanto como uma anonimização especializada para algum termo-alvo. Disponibiliza-se assim opções para serem anonimizados os (1) nomes, (2) documentos e (3) endereços. Estas opções podem ser utilizadas em conjunto, mediante a necessidade do utilizador. O programa está disponível no [PyPI](#) e pode ser instalado através do comando `pip install hshield`.

```
Data anonymizer tool

positional arguments:
  filename

optional arguments:
  -h, --help            show this help message and exit
  -n, --name            anonymize only names
  -d, --document        anonymize only documents
  -a, --address          anonymize only addresses
  -o OUTPUT, --output OUTPUT
                        output file

Build by Henrique, José and Alex
```

Modos de uso do Hshield



4 Conclusão

O Hshield foi apresentado como um *software* de anonimização de textos em termos de três módulos, cada um responsável por aspectos diferentes de conteúdos do texto. Ora, nada impede da existência de mais módulos, com mais tipo de informação a ser anonimizada. Deste modo, como trabalho futuro, pretende-se a expansão do programa para abranger mais conteúdos de um dado texto. Para além disto, visiona-se também a expansão da ferramenta para dar suporte a mais linguagens para além do português.

No que respeita a anonimização dos nomes, foi mostrado o cuidado tomado para manter a coesão semântica das frases. No entanto, ainda há um factor por melhorar. Este factor é quando há a identificação de entidades com nomes diferentes que na verdade correspondem a uma só entidade. Isto é, se num momento uma entidade "José Pedro" for tratada com estes dois nomes mas, noutro momento, ser tratada somente como "José", haverá uma diferenciabilidade na identificação destas entidades. Como trabalho futuro, procurar-se-á adicionar novas *features* para lidarem com esta questão.

Referências

- [1] Universidade de Coimbra. Proteção de dados pessoais: Anonimização e pseudonimação. Online, data de acesso.
- [2] SERPRO. Dados anonimizados. Online, 8 de Jun de 2023.
- [3] Abraão Almeida. O que são dados anonimizados e como realizar esse processo? Online, 8 de Jun de 2023.