UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Aperfeiçoamento de Tecnologias de Anonimização para o Contexto de Dados Não Estruturados

Raquel Alexandra Moleira Domingos

Mestrado em Engenharia Informática

Trabalho de Projeto orientado por:
Prof. Doutor Diogo Filipe Marques Soares

2025

aos meus pais, ao meu irmão e ao meu tio.

Agradecimentos

Gostava de expressar a minha gratidão a todos aqueles que me apoiaram e me motivaram durante a realização desta tese.

Em primeiro lugar, quero agradecer ao professor Diogo Soares, que me orientou durante este trabalho, pela disponibilidade que sempre demonstrou para comigo, assim como pela sua compreensão e pelos seus conselhos.

Também estou muito grata à Trust Systems e a todos os seus elementos, que me acolheram extremamente bem e me ensinaram bastante ao longo destes meses. Agradeço pela orientação à Ana Guimarães e ao Duarte Olival, pois apoiaram-me e, graças à sua ajuda, consegui ultrapassar as dificuldades que surgiram ao longo da tese.

Por fim, quero agradecer à minha família e amigos, que me apoiam sempre nas minhas escolhas e me motivam todos os dias a ser uma pessoa melhor.

Resumo

A circulação de dados, especialmente dados não estruturados, ocupa um papel primordial nas tecnologias atuais. Contudo, é essencial o seu tratamento, caso contrário, poderá ser posta em causa a privacidade de muitos sujeitos. Por este motivo, deve ser possível a anonimização de todos estes dados, de forma a equilibrar o nível de privacidade e de utilidade dos mesmos. A Trust Systems, empresa dedicada à segurança de informação, já possui algum envolvimento na área de anonimização com o desenvolvimento do projeto HYDE, possuindo este a funcionalidade de anonimizar dados estruturados, e já tendo realizado algumas teses e pesquisas sobre a anonimização de dados não estruturados.

Esta tese pretende aprofundar a anonimização de dados não estruturados, criando um sistema de anonimização de PDFs em português para ser integrado no projeto HYDE. Este sistema permite a deteção dos termos a anonimizar através de um modelo NER com a arquitetura Transformer, treinado com vários corpora de idioma português no formato IOB, acabando por obter na avaliação final do sistema os valores 0.82, 0.58 e 0.65 como média de Recall, Precisão e F1-Score, respetivamente. O sistema também possuí uma Aplicação Java responsável pela comunicação do modelo com o projeto HYDE, assim como por construir o PDF anonimizado final.

Palavras-chave: Anonimização; Dados Não Estruturados; Privacidade; *Named Entity Recognition*; *Natural Language Processing*

Abstract

The circulation of data, especially unstructured data, plays a primary role in current technologies. However, its processing is essential, otherwise, the privacy of many individuals may be put at risk. Therefore, it must be possible to anonymize all this data in order to balance their levels of privacy and utility. Trust Systems, a company dedicated to information security, is already involved in the area of anonymization with the development of the project HYDE, which offers the functionality to anonymize structured data, and by having already developed some thesis and researches about the anonymization of unstructured data.

This thesis focuses on further exploring the anonymization of unstructured data by creating a system that anonymizes portuguese PDFs to be integrated into project HYDE. This system allows the detection of terms to be anonymized using a NER model with a Transformer architecture, trained with various datasets in the portuguese language and in IOB format, obtaining final evaluation scores of 0.82, 0.58 and 0.65 as the average of recall, precision and F1-Score, respectively. The system also contains a Java Application, responsible for the communication between the model and project HYDE, as well as for the construction of the final anonymized PDF.

Keywords: *Anonymization*; *Unstructured Data*; *Privacy*; *Named Entity Recognition*; *Natural Language Processing*

Conteúdo

Lista de Figuras

Lista de Tabelas

Listings

Capítulo 1

Introdução

1.1 Motivação

Atualmente, grande parte dos sistemas recorrem à recolha e análise de vários dados, com o objetivo de melhorarem os seus serviços. Por exemplo, a digitalização de documentos judiciais ajuda a tomada de decisões em tribunais [19]. Contudo, a circulação pública destes dados e o acesso a mais informação do que o necessário comprometem os dados sensíveis dos indivíduos referenciados. Sendo assim, e com o objetivo de respeitar o direito à privacidade estabelecido na Declaração Universal dos Direitos Humanos e as medidas estabelecidas pelo Regulamento Geral sobre a Proteção de Dados (RGPD), é necessário a obtenção de consentimento prévio do titular dos dados para que esta informação possa ser partilhada. Todavia, este consentimento é difícil de obter para todos os utilizadores, sendo mais prático recorrer a técnicas de anonimização sobre os dados [33].

A maioria dos dados, atualmente, existem sob a forma de dados não estruturados [63], correspondendo sobretudo a texto. Este tipo de dados traz certos desafios que dificultam a criação de uma tecnologia capaz de identificar qual a informação que deve ser anonimizada. Com dados não estruturados, é necessária a noção do contexto de cada termo, uma vez que certos termos poderão não ser considerados para anonimização se isolados, mas o mesmo não acontece se o termo for enquadrado em contextos específicos, como, por exemplo, apenas a palavra "jardim" não deve ser considerada para anonimização, no entanto, caso essa palavra esteja inserida num nome, tal como "João Jardim", então já deve ser vista como um termo a anonimizar. Outra dificuldade assenta sobre o domínio dos dados de treino, ou seja, se um modelo for treinado para identificar termos sensíveis num contexto legal, a sua tarefa pode ser bem realizada num documento de tribunal, mas apresentar má performance se for analisar um texto vindo das redes sociais. Sendo assim, é importante definir um domínio que seja o foco da anonimização e treinar um modelo com dados de treino adequados para o mesmo. Para além disso, deve-se ter em atenção que os dados de treino também devem conter as *labels* que se pretende identificar, por exemplo, as *labels* PESSOA e LOCAL, e não incluir as *labels* que não serão necessárias identificar, tal como a *label* JURISDICAO se o foco não for documentos de domínio legal. Por fim, a falta de recursos computacionais que permitem o processamento de modelos complexos impede que a anonimização de dados não estruturados seja um objetivo simples e rápido, portanto, existe a necessidade de criação de *frameworks* e técnicas

de anonimização relativas a dados não estruturados. Para que esta anonimização seja mais eficaz, em prática, a mesma deve ser automática, já que a sua realização de forma manual consome muito tempo [33].

Para responder aos desafios mencionados anteriormente, surge este projeto no contexto da empresa Trust Systems [7], a qual tem como principal objetivo o desenvolvimento e oferta de soluções relacionadas com a segurança de informação. A empresa Trust Systems atua desde 2016 nesta área e é uma das empresas pertencentes ao ecossistema Inoweiser.

Visto que o seu foco é a segurança de informação, a Trust Systems tem como uma das suas preocupações o problema da circulação de dados sensíveis, e consequentemente, a quebra de privacidade de indivíduos. A empresa tem como objetivo combater estes problemas, tendo criado para tal o projeto HYDE, cuja função consiste na anonimização de dados, tendo sempre em conta a consistência (quando escolhido um tipo de anonimização que o permita), privacidade e utilidade dos dados.

O projeto HYDE é um projeto desenvolvido pela empresa Trust Systems que consiste numa plataforma que permite anonimização de dados. No entanto, até à realização desta tese, apenas se encontrava implementada a anonimização de dados estruturados, pretendendo-se apronfundar esforços anteriormente realizados, durante a concretização da tese do Luís Santos [24], de forma a adicionar ao projeto a funcionalidade de anonimização de dados não estruturados, algo de extrema importância, já que, como referido anteriormente, a maior parte dos dados acabam por ser não estruturados. O projeto HYDE não proporciona a anonimização de dados estruturados apenas para português, mas também para inglês, francês e alemão. Para a anonimização de dados estruturados, são identificados os elementos sensíveis dos dados fornecidos ao projeto HYDE, recorrendo a dicionários e reconhecimento de padrões. Os dados identificados são então mostrados ao utilizador, de forma a que este possa concordar ou discordar com os mesmos para anonimização, e também decidir qual a técnica de anonimização a utilizar para cada um deles. O projeto HYDE apenas implementa as técnicas de Pseudonimização, *Masking*, Supressão, e uma versão de Pseudonimização sem consistência, apesar de haver intenções de virem a ser implementados mais métodos no futuro. Posteriormente, é criada uma cópia da base de dados, pois esta é aquela que é anonimizada. Adicionalmente, a consistência dos dados também é garantida quando estes passam a ser anonimizados, ou seja, quando se realizar a pseudonimização de algum termo em específico, todas as ocorrências deste termo são anonimizadas com o mesmo pseudónimo.

1.2 Objetivos

Este trabalho assenta na aprimoração do projeto HYDE. O seu objetivo é desenvolver um sistema de anonimização, a ser integrado no projeto HYDE, que adicione ao mesmo a capacidade de anonimizar também dados não estruturados, mais especificamente, ficheiros de texto em formato PDF. Este sistema deve incluir um modelo que realize tarefas *Named Entity Recognition* (NER), identificando os termos a anonimizar no texto do PDF, e uma aplicação, PDFAnonymizer, que seja

responsável pelo processamento do PDF, assim como pela construção do PDF final anonimizado. O modelo também deve ser inserido numa aplicação Flask de forma a que possa ser acedido pelo PDFAnonymizer. O modelo deve ser capaz de identificar os termos a anonimizar em documentos em língua portuguesa, e a anonimização deve ser realizada com recurso ao método Supressão. O domínio dos documentos em que o modelo se irá focar é o domínio geral.

1.3 Contribuições

Este trabalho permitiu realizar as seguintes contribuições de melhoria ao Projeto HYDE:

- Obtenção de um modelo NER com arquitetura Transformer, o qual é capaz de identificar termos classificados como Organização, Pessoa e Local.
- Aplicação Flask com o modelo obtido, e com a adição de uma componente regex, que permite identificar termos classificados como Email, Telefone, Código Postal e Número de Identificação Nacional.
- Aplicação em Java que é responsável pela comunicação entre o projeto HYDE e o modelo e pela reconstrução do PDF, já com os termos identificados anonimizados pela técnica de Supressão.

1.4 Estrutura do documento

O resto deste documento está organizado da seguinte forma:

- **Capítulo 2** – apresenta os conceitos básicos para o entendimento do assunto em investigação, incluindo a explicação dos vários tipos de dados que identificam um sujeito e dos variados métodos de anonimização. Também aborda as contribuições já realizadas sobre o tópico em questão, assim como a introdução a métodos e ferramentas que foram utilizados durante a tese.
- **Capítulo 3** - descreve todo o procedimento até à seleção dos melhores modelos NER. Inclui a explicação do pré-processamento realizado dos corpora utilizados, assim como o esclarecimento de como foram obtidos os resultados que permitiram a seleção dos modelos.
- **Capítulo 4** - refere todo o processo de implementação das aplicações onde correm os modelos NER escolhidos, e da aplicação responsável pela reconstrução do documento e pela comunicação com projeto HYDE. Para além disso, é explicada a infraestrutura do sistema e os passos necessários para completar a integração com o projeto HYDE.
- **Capítulo 5** - apresenta o método de avaliação final, referindo qual foi o corpus utilizado e mostrando os resultados obtidos, assim como as conclusões obtidas destes.

- **Capítulo 6** - conclusão da tese, referindo o que era pretendido, o que foi feito e o que foi concluído. Para além disso, menciona as melhorias que podem ser realizadas no futuro.

Capítulo 2

Conceitos e Trabalho relacionado

Este capítulo inclui algumas definições e conceitos necessários para um melhor entendimento do problema a abordar, assim como refere ferramentas e contribuições prévias da literatura.

2.1 Conceitos

2.1.1 Dados Estruturados e Dados Não Estruturados

Com a crescente utilização da tecnologia, existem cada vez mais dados a serem recolhidos e consultados, sendo estes organizados e usados de diferentes formas. Os dados podem ser estruturados, e neste caso, são armazenados de acordo com um formato específico, por exemplo, tabelas de dados. Por outro lado, também existem dados sem um formato pré-estabelecido, como documentos de texto, imagens ou ficheiros de áudio. Estes são denominados como dados não estruturados [63].

2.1.2 Anonimização

A anonimização de dados consiste na modificação de informação de forma a que esta não possa ser associada a nenhum indivíduo. Os dados a anonimizar podem ser divididos em três categorias: identificadores diretos, quasi-identificadores e atributos sensíveis [33].

Identificadores Diretos são aqueles que devem ser sempre anonimizados, visto que são únicos para cada indivíduo. Alguns exemplos deste tipo de dados são NIFs, números de telefone e moradas.

Quasi-identificadores referem-se a dados que, quando considerados isoladamente, não permitem a identificação de um sujeito. No entanto, a sua combinação já o possibilita. Estes dados incluem datas de nascimento, género, código postal, entre outros.

Atributos sensíveis são atributos que não devem ser revelados, tais como crenças religiosas ou dados de saúde.

A anonimização não deve ser confundida com de-identificação, a qual se foca apenas em remover identificadores diretos [33], podendo o indivíduo ainda ser identificado devido à combinação de quasi-identificadores.

2.1.3 Identificação de Termos a Anonimizar

Uma das fases essenciais para a anonimização de dados assenta sobre a identificação da informação que deve ser anonimizada. No caso de dados que tenham um formato definido, como por exemplo emails, esta identificação pode ser realizada recorrendo à verificação de padrões. Outra forma de identificar dados poderá ser a utilização de dicionários, por exemplo, utilizar uma lista de nomes próprios para identificar se um termo pode ou não ser considerado como o nome de uma pessoa. Estes tipos de identificação funcionam para dados estruturados, mas relativamente a dados não estruturados, o mesmo não se pode afirmar, pois estes procedimentos não têm em conta o contexto das palavras. Para que seja possível considerar a relação entre as várias palavras de um documento de texto, é necessário recorrer a técnicas de Aprendizagem Automática (AA) [47].

Um dos processos mais conhecidos para executar esta identificação é o *Named Entity Recognition* (NER). NER é a tarefa que reconhece num documento entidades, ou seja, identificadores que podem dizer respeito, por exemplo, a pessoas, locais, organizações, e as quais classifica em categorias pré-definidas, tais como PERSON, LOCAL, ORGANIZATION. A tarefa NER pode ser realizada usando várias abordagens, tais como, abordagem baseada em regras, aprendizagem não supervisionada e aprendizagem supervisionada [30].

Abordagem baseada em regras, que tal como o nome indica, classifica informação de acordo com regras, as quais são definidas por especialistas. Desta forma, o processo torna-se mais eficiente e preciso. Existem diversos sistemas que utilizam esta abordagem como FASTUS [28], o qual funciona melhor quando apenas uma pequena quantidade de texto em análise é relevante, LaSie [29] que extrai informação de vários domínios e Facile [15] que categoriza um texto em diversos idiomas. Contudo, a abordagem baseada em regras é restrita a um só domínio, e necessita de intervenção humana para a definição das regras. Para além disso, e tal como mencionado anteriormente, este processo de identificação não é ideal para dados não estruturados, uma vez que não considera a relação entre as palavras [30].

Aprendizagem não supervisionada é um método de AA. Este tipo de abordagem utiliza técnicas como *clustering* para lidar com dados não anotados. O facto de não utilizar corpora com dados classificados, torna esta aprendizagem mais adaptável e, consequentemente, adequada para classificar dados que impliquem mais do que um domínio. Existem exemplos de aplicações desta abordagem em diferentes domínios, por exemplo, no domínio das redes sociais com o trabalho realizado por Liu e Zhou [35], o qual utiliza a redundância de *tweets* para realizar NER em duas fases com *tweets* de conteúdo semelhante, e no domínio biomédico com o método de Zhang e Elhadad [66], que em vez de recorrer a regras ou dados de treino anotados, utiliza terminologias,

sintaxes superficiais e estatísticas de corpus para efetuar as tarefas NER. Contudo, esta abordagem tem como desvantagem a dificuldade da sua avaliação face à avaliação da aprendizagem supervisionada, uma vez que não pode utilizar dados anotados e, consequentemente, também não pode utilizar métricas como *f1-score* ou *accuracy* visto que não existem *labels standard* com as quais se pode comparar o *output* [30].

Aprendizagem supervisionada também é indicada para o contexto de dados não estruturados, podendo serem usados variados tipos de algoritmos de AA para concretizar a identificação dos termos no texto. Esta abordagem foca-se em utilizar dados categorizados para o treino e avaliação de modelos. Existem diversos algoritmos que usam Aprendizagem Supervisionada, nomeadamente, o modelo de Hidden Markov [67], Conditional Random Fields [45] e Support Vector Machines [52]. No entanto, a utilização desta abordagem requer grandes quantidades de dados anotados por especialistas, o que significa um elevado consumo de tempo e dinheiro [30].

2.1.4 Técnicas de Anonimização

Após a identificação dos dados que devem ser anonimizados, é necessário selecionar técnicas para o concretizar. Consideremos as seguintes técnicas: generalização, supressão, pseudonimização, *swapping*, *masking* e distorção. Os exemplos apresentados em seguida, consideram os dados originais apresentados na figura 2.1, retirada do estudo de 2019 realizado por Murthy et al. [41].

ID	Name	Address	Postcode	Usage (kW)
123	Alice	117, Jalan Kinrara 1	35400	434
234	Bob	14, Jalan Presint 9/1	51400	289
456	John	23, Jalan Amanah 5	81200	45
789	Sarah	89, Jalan Nuri 8	68100	872

Figura 2.1: Dados originais [41]

Generalização consiste na substituição dos dados a anonimizar por termos mais gerais, levando a que o leque de indivíduos associados à informação em causa aumente, dificultando a identificação de sujeitos. De forma a exemplificar, considera-se a figura 2.2, que generaliza os kW utilizados para intervalos de valores.

ID	Name	Address	Postcode	Usage (kW)
123	Alice	Jalan Kinrara 1	35400	400 - 450
234	Bob	Jalan Presint 9/1	51400	250 - 300
456	John	Jalan Amanah 5	81200	1 - 50
789	Sarah	Jalan Nuri 8	68100	850 - 900

Figura 2.2: Dados após generalização da coluna *Usage*

Supressão elimina completamente os dados a anonimizar, substituindo-os com outros valores sem significado, como "∗" [41]. Na tabela apresentada na figura 2.3, podem ser observadas as colunas ID e *Name* após ser aplicada a supressão.

ID	Name	Address	Postcode	Usage (kW)
***	***	117, Jalan Kinrara 1	35400	434
***	***	14, Jalan Presint 9/1	51400	289
***	***	23, Jalan Amanah 5	81200	45
***	***	89, Jalan Nuri 8	68100	872

Figura 2.3: Dados após supressão das colunas ID e *Name*

Pseudonimização trata-se da substituição de dados por pseudónimos. Um exemplo seria o nome "Miguel" que poderia ser anonimizado ao ser substituído pelo nome "Pedro". Este método pode também ser aplicado de forma consistente, ou seja, certos dados seriam associados a pseudónimos específicos, contudo, esta associação teria de ser armazenada em separado e em segurança [33].

Swapping redistribui aleatoriamente os valores de uma coluna. Por exemplo, na figura 2.4, apresenta-se uma nova distribuição dos valores da coluna *Name*, diferente da ordem original (presente na figura 2.1).

ID	Name	Address	Postcode	Usage (kW)
123	Bob	117, Jalan Kinrara 1	35400	434
234	Sarah	14, Jalan Presint 9/1	51400	289
456	Alice	23, Jalan Amanah 5	81200	45
789	John	89, Jalan Nuri 8	68100	872

Figura 2.4: Dados após *swapping* da coluna *Name*

Masking envolve a alteração de certos caracteres dos valores das colunas por outros caracteres. Por exemplo, o código postal 1234-567 pode ser alterado para o valor 1234-∗ ∗ ∗ [41]. Na figura 2.5 também é possível observar que foi aplicada a técnica *masking* nos valores da coluna *Postcode*, sendo revelado apenas o primeiro dígito destes valores.

ID	Name	Address	Postcode	Usage (kW)
123	Alice	117, Jalan Kinrara 1	3XXXX	434
234	Bob	14, Jalan Presint 9/1	5XXXX	289
456	John	23, Jalan Amanah 5	8XXXX	45
789	Sarah	89, Jalan Nuri 8	6XXXX	872

Figura 2.5: Dados após *masking* da coluna *Postcode*

Distorção altera a informação para outros valores, podendo mais tarde estes serem revertidos para o estado original. Certos atributos podem ser alterados de acordo com a expressão $Vd = Vu + Vr$, correspondendo Vu aos dados originais que são adicionados a outros dados Vr, resultando em Vd, os dados distorcidos [41]. Na figura 2.6 encontram-se distorcidos os dados pertencentes à coluna *Postcode*.

ID	Name	Address	Postcode	Usage (kW)
123	Alice	117, Jalan Kinrara 1	53cc586259ac045be913eb94c895507e	434
234	Bob	14, Jalan Presint 9/1	976923868f1f8c52da095843c9d6e98	289
456	John	23, Jalan Amanah 5	a9340bcc37991d716ba1ccc40c5c9c62	45
789	Sarah	89, Jalan Nuri 8	53ea609905a3df24d569515184ffcf0d	872

Figura 2.6: Dados após distorção da coluna *Postcode*

Algo que se deve ter em atenção quando se decide a técnica de anonimização a usar é que existe um *trade-off* entre o nível de privacidade dos dados e a utilidade destes [33]. Algumas técnicas podem anonimizar bem os dados, mas remover grande parte do seu significado, enquanto que outras podem preservar a maioria do conteúdo dos dados, mas não a privacidade da informação. A figura 2.7 apresenta os níveis de utilidade e privacidade para cada uma das técnicas de anonimização antes descritas.

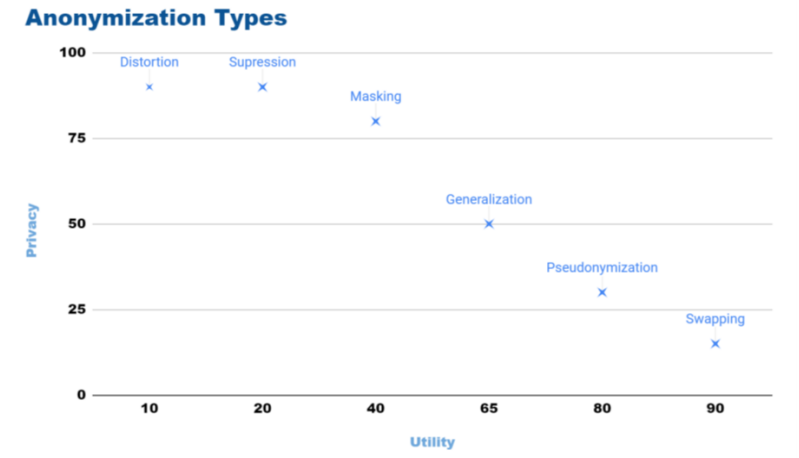

Figura 2.7: Gráfico da relação entre privacidade e utilidade [22]

2.2 Trabalho Relacionado

2.2.1 Abordagens NLP

As abordagens *Natural Language Processing* (NLP) [33] focam-se, maioritariamente, em métodos de de-identificação e de ofuscação. Relativamente à de-identificação, grande parte da investigação é relacionada com a área clínica de NLP, com o objetivo de detetar *Protected Health Information* (PHI) em documentos de texto. Para atingir este objetivo, usam-se técnicas baseadas em regras e de AA, porém, duas das grandes dificuldades encontradas foram a falta de textos anotados e a falta de normas de anotação universais para PHI, tornando mais difícil a passagem de dados para outros domínios. Para além disso, e como anteriormente referido na secção 2.1.2, a de-identificação apenas trata de identificadores diretos, não sendo completamente obtida a privacidade do sujeito em questão, visto que o mesmo pode ser identificado com base nos seus quasi-identificadores.

Os métodos de ofuscação pretendem detetar e esconder quasi-identificadores, baseando-se em textos provenientes de redes sociais. Estes métodos não são capazes de transformar o texto original, apenas executando alterações a nível interno (nível de representação vetorial latente), não sendo possível esconder totalmente a identidade do sujeito.

Adicionalmente, estas técnicas também podem eliminar demasiada informação, visto que todas as ocorrências detetadas serão tratadas sem ter em causa o impacto que estas remoções poderão vir a ter na utilidade da informação.

2.2.2 Ferramentas NER

Uma parte importante neste trabalho é conhecer as ferramentas que permitem a extração e identificação de informação de um texto. Ferramentas estas que facilitam o treino, validação e testes que levam à obtenção de um modelo NER.

O estudo da autoria de Jehangir et. al [30] indica algumas ferramentas NER, nomeadamente as ferramentas Spacy[1], NLTK[2] e Apache openNLP[3]. De acordo com Schmitt et. al [51], que tinha como objetivo a comparação entre as ferramentas anteriormente mencionadas e as ferramentas Gate[4] e Stanford CoreNLP[5], a última apresenta uma melhor performance relativamente às restantes, as quais possuem um desempenho semelhante entre si. Contudo, vale referir que Stanford CoreNLP tem uma melhor performance relativamente ao corpus CoNLL-2003 [57], uma vez que foi parcialmente treinada com o mesmo. Já o corpus GMB [16], também utilizado no estudo, apresentou uma performance melhor do que as outras ferramentas, no entanto, com uma menor diferença.

[1] `https://spacy.io/`
[2] `https://www.nltk.org/`
[3] `https://opennlp.apache.org/`
[4] `https://gate.ac.uk/`
[5] `https://stanfordnlp.github.io/CoreNLP/`

2.2.3 Conditional Random Field

Conditional Random Field (CRF) é um tipo de modelo probabilístico cuja arquitetura se baseia em autómatos probabilísticos de estado finito. Utiliza *labels* anteriormente previstas e o contexto de cada palavra para calcular a probabilidade de correspondência com uma *label* e escolher aquela com máxima probabilidade [26] [45].

Os dados que são fornecidos ao modelo são convertidos num vetor de parâmetros para cada palavra, o qual inclui o contexto da palavra em questão. Durante o treino de um modelo com arquitetura CRF, as funções de probabilidade de cada *label* são ajustadas de acordo com os vetores de parâmetros de dados de treino corretamente anotados, de forma a maximizar a correspondência entre um termo e a sua classificação [45].

Esta arquitetura é uma opção que é utilizada durante a seleção de modelos, já que a sua consideração pelo contexto e pelas probabilidades das *labels* calculadas anteriormente a tornam indicada para lidar com dados não estruturados e com tarefas NER [45].

2.2.4 Transformers

Quando se fala em *Deep Learning*, as *Recurrent Neural Networks* (RNNs) representavam os modelos estado de arte relativamente a tarefas sequenciais, como é o caso de NER. No entanto, estes modelos possuem algumas limitações no que toca à paralelização e eficiência. Vaswani et. al [58] apresentaram então o Transformer, um modelo completamente baseado em mecanismos de atenção, os quais permitem capturar relações entre partes do texto a qualquer distância, possibilitando assim uma maior paralelização.

A arquitetura do Transformer, representada na figura 2.8 inclui um *encoder* e um *decoder*, cada um sendo composto por 6 camadas de redes neuronais.

O *encoder* é responsável por receber uma sequência como *input*, e processá-la de forma a obter uma sequência de representações contínuas. Cada camada do *encoder* possui 2 sub-camadas: uma com um mecanismo de *multi-head self-attention* e a outra com redes *feed-forward*. O mecanismo *multi-head self-attention* permite consultar várias representações anteriores, calculando em paralelo a atenção que deve haver entre partes do texto a diferentes distâncias na representação, e acabando por concatenar os resultados. A segunda sub-camada possui uma rede neuronal completamente conectada e *feed-forward*, ou seja, uma rede que atua independentemente para cada posição, aplicando transformações lineares e uma ativação ReLU.

O *decoder* gera a sequência *output* de *tokens* baseando-se no *output* do *encoder* e também em *tokens* previamente gerados. O *decoder* apresenta as duas sub-camadas referidas no *encoder*, mas também uma terceira sub-camada que aplica o mecanismo de *multi-head attention* ao *output* do *encoder*. A sub-camada *self-attention* do *decoder* é mascarada, ou seja, cada posição pode apenas ver os *outputs* das posições anteriores.

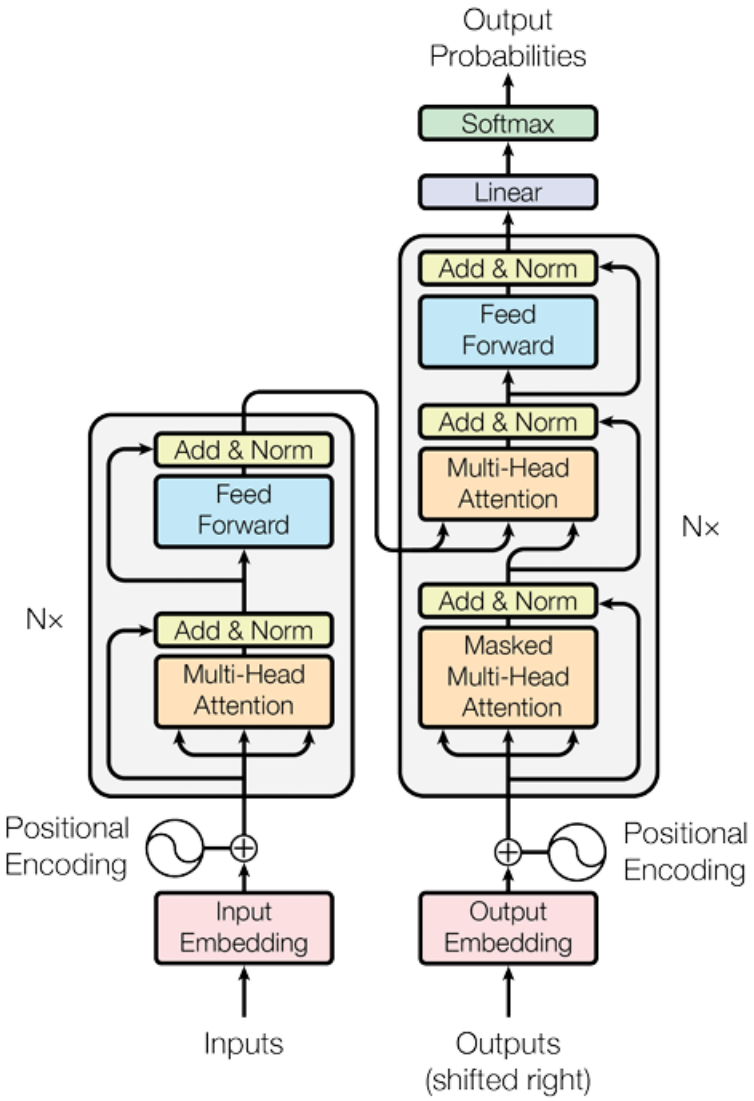

Figura 2.8: Arquitetura do Transformer [58]

2.2.5 Large Language Models

Um *Large Language Model* (LLM) é um modelo treinado com extensas quantidades de dados e contextos de aprendizagem, para que o mesmo seja capaz de entender e processar a linguagem humana [64].

Tem-se observado um aumento na relevância de LLMs, comparativamente a ferramentas como Microsoft Presidio[6] e Azure Language Services[7], as quais são vistas como cada vez mais limitadas na área de anonimização. As vantagens de LLMs face a estas ferramentas incluem o alcance de um melhor balanço entre a privacidade e utilidade dos dados, algo não considerado por muitas das restantes ferramentas de anonimização atuais, e também o facto das LLMs atingirem quase o nível de performance humano no que diz respeito a identificação de atributos num texto [55].

As LLMs mais avançadas incluem o GPT-5 [44], desenvolvido pela OpenAI, BERT [21], desenvolvido pela Google, LLaMa 3 [40], desenvolvido pela Meta e ainda o modelo Qwen [13]. Porém, apenas os últimos três modelos são *open-source* [17].

Staab et. al [55] desenvolveram uma *framework* que consiste em dois LLMs, um dedicado à realização da anonimização de texto e o outro responsável por inferir atributos presentes no texto. O resultado do LLM de anonimização é fornecido ao LLM de inferência, para que este tente identificar atributos no texto anonimizado. O resultado da inferência é então devolvido ao modelo de anonimização, de forma a que este consiga melhorar o nível de privacidade do texto. Este processo repete-se até o modelo de inferência não ser capaz de identificar mais atributos ou até ser atingido um dado número de iterações.

No entanto, deve-se ter em conta que a utilização de uma LLM remota no sistema implementado durante esta tese, levaria ao comprometimento dos dados que lhe são dados. Desta forma, foram apenas fornecidos dados *open-source* durante a utilização de modelos remotos.

2.2.6 Corpora NER

Para realizar a identificação dos termos a anonimizar, são necessários dados que sejam categorizados e apresentados conforme Inside-outside-beginning (IOB). O IOB é um sistema de marcação de *tokens*, cujo objetivo é identificar o tipo e posição destes numa entidade [34]. Assim, será associada a letra "B" aos *tokens* que iniciam a classificação de um termo, será associado "I" aos *tokens* que constituem a parte intermédia da classificação de um termo, e associa-se o "O" a todos os *tokens* que não são classificados [33]. De forma a exemplificar, é apresentada a figura 2.9 [37].

[6] `https://microsoft.github.io/presidio/`
[7] `https://azure.microsoft.com/en-us/products/ai-services/ai-language`

A	O
falta	O
de	O
intervenção	O
do	O
Ministério	B-ORGANIZACAO
Público	I-ORGANIZACAO
nas	O
ações	O
em	O
que	O
deva	O
figurar	O
como	O
fiscal	O
da	O
lei	O
e	O
da	O
Constituição	B-LEGISLACAO
(	O
custus	O
legis	O
et	O
constituitionis	O
’	O
)	O

Figura 2.9: Excerto do corpus de treino de LeNER-BR

Numa pesquisa publicada em 2023 [30], são apresentados alguns corpora que foram colocados em hipótese para treinar o modelo NER, tais como CoNLL-2002 [56] com foco na língua neerlandesa, CoNLL-2003 [57] que incluí as línguas alemã e inglesa e OntoNotes [62] que se foca em árabe, inglês e chinês. Também considerámos pertinentes os corpora WikiGold [43], WikiCoref [27] e HYENA [65], os quais foram criados a partir de artigos da Wikipedia inglesa. Para além disso, também são apresentados WNUT-2017 [20], NCBI disease corpus [23], BioCreative V chemical disease relation [61] e Genia corpus [32], os quais não foram considerados, uma vez que WNUT-2017 se baseia no estilo de escrita da rede social Twitter (conhecida atualmente como X), e os restantes consistem em dados biomédicos ou químicos, sendo limitados a domínios específicos.

Wang et. al [59] utilizou o corpus MultiCoNER, introduzido por Malmasi et. al [39], com o objetivo de desenvolverem um sistema *Knowledge-based* que possa ser aplicado a um NER de vários idiomas. Este corpus inclui várias línguas presentes no projeto HYDE, tais como inglês, alemão, francês, português e espanhol.

A maioria dos corpora mencionados anteriormente focam-se bastante no idioma inglês, havendo necessidade de descobrir outros corpora que abordem também as restantes línguas. Especificamente, para a língua portuguesa existem dois corpora: LeNER-Br [37] e Harem [50] (e a sua versão mais recente Second Harem [25]). Adicionalmente, existe ainda mais um corpus para o idioma português, desenvolvido durante a tese do Luís Santos [24], o LexPT. O LexPT é constituído por documentos do domínio legal e é composto por dois corpora: o LexPT-H, que consiste em dados anotados manualmente, e o LexPT-GPT, que se refere a dados gerados pelo ChatGPT.

Para o alemão existe um corpus baseado em vários artigos da Wikipédia alemã e que foi usado

por Bervoka et. al [14]. O estudo realizado por Wang et. al [60] menciona dois corpora também de interesse para este trabalho: o corpus AnCora [1] que inclui as linguas espanhol e catalão, e o corpus TCA KBP 2017 [2], que aborda o espanhol, o inglês e o mandarim.

Os corpora que incluem o idioma luxemburguês são escassos, tendo sido apenas encontrados dois corpora: Wikiann [48] e LuxemBERT [36]. No entanto, estes não estão formatados de acordo com o método IOB, tendo de ser convertidos para tal. O corpus Wikiann [48] inclui todos os idiomas mencionados nesta tese.

Na tabela 2.1, estão documentados os vários corpora apresentados, assim como os idiomas que abordam.

Tabela 2.1: Corpora e idiomas abordados. Os idiomas apresentados incluem português (PT), inglês (EN), francês (FR), alemão (DE), espanhol (ES), holandês (NL) e luxemburguês (LB).

	PT	EN	FR	DE	ES	NL	LB
CoNLL-2002						✓	
CoNLL-2003		✓		✓			
WikiGold		✓					
WikiCoref		✓					
HYENA		✓					
MultiCoNER	✓	✓	✓	✓	✓	✓	
LeNER-Br	✓						
Harem	✓						
LexPT	✓						
Germeval2014				✓			
AnCora					✓		
TAC KBP 2017		✓			✓		
LuxemBERT							✓
Wikiann	✓	✓	✓	✓	✓	✓	✓

2.2.7 Técnicas PPDP

Privacy-Preserving Data Publishing (PPDP) consiste em técnicas para a publicação de dados sem violar a privacidade de qualquer indivíduo, tendo como objetivo não só identificar e anonimizar os identificadores diretos, como também os quasi-identificadores.

A técnica PPDP mais utilizada é k-anonimidade, a qual será explicada em seguida. Consideremos uma base de dados, com vários registos, cada um pertencente a uma dada entidade, sendo que uma entidade diz respeito a um indivíduo ou objeto. Cada uma dessas entidades tem um conjunto de atributos, os quais podem incluir quasi-identificadores. Diz-se que um conjunto de dados possui k-anonimidade se para cada combinação de quasi-identificadores existem, pelo menos, k entradas na tabela. Um exemplo poderá ser observado na figura 2.10 [22], onde os quasi-identificadores são as colunas *Age* e ZIP, sendo possível notar que nenhuma das combinações existentes entre estes dois atributos é exclusiva, havendo sempre, pelo menos, duas combinações iguais. Desta forma,

podemos afirmar que este conjunto de dados anonimizado possui 2-anonimidade.

	QI_1	QI_2	S_1
ID	Age	Zip	Disease
1	5	15	Flu
2	15	25	Fever
3	28	28	Diarrhea
4	25	15	Fever
5	22	28	Flu
6	32	35	Fever
7	38	32	Flu
8	35	25	Diarrhea

	QI_1	QI_2	S_1
ID	Age	Zip	Disease
1	0-20	10-30	Flu
2	0-20	10-30	Fever
3	20-30	10-30	Diarrhea
4	20-30	10-30	Fever
5	20-30	10-30	Flu
6	30-40	20-40	Fever
7	30-40	20-40	Flu
8	30-40	20-40	Diarrhea

Figura 2.10: Dados com 2-anonimidade

Para além de k-anonimidade, existe ainda a ℓ-diversidade [38], a qual considera que para cada classe de equivalência, ou seja, conjunto de entidades com os mesmos valores de quasi-identificadores, existem, pelo menos, ℓ valores de atributos sensíveis diferentes. Por exemplo, na figura 2.11, todas as entradas da tabela pertencem à mesma classe de equivalência, visto que têm os mesmos valores de quasi-identificadores (colunas *Age* e *Country*). Como esta classe de equivalência possui dois valores diferentes na coluna *Political views*, a qual representa o atributo sensível, então este corpus apresenta 2-diversidade.

	Quasi Identifiers		Sensitive Attribute
ECs	Age	Country	Political views
C_1	35-37	North America	Liberal
	35-37	North America	Conservative
	35-37	North America	Conservative
	35-37	North America	Conservative

Figura 2.11: Dados com 2-diversidade

Apesar da grande utilização dos métodos k-anonimidade e ℓ-diversidade, os mesmos assumem que os dados são estruturados, sendo este o foco da maioria dos estudos das técnicas PPDP [33].

No entanto, se num trabalho futuro for possível transformar os dados não estruturados em dados estruturados, poder-se-ia utilizar k-anonimidade ou ℓ-diversidade. Em 2009, Gardner et. al [26], depois de identificarem os atributos sensíveis do texto, realizam uma fase iterativa composta por dois passos: a extração dos atributos e a sua ligação ou adição a novas ou já existentes entidades da base de dados. Abordagens mais recentes têm sido propostas como Text2Struct [18], uma *pipeline* que se baseia na anotação de termos e relações num texto, e MedPromptExtract [54], uma *framework* que pretende extrair dados estruturados de um texto, preservando a sua confidencialidade.

As técnicas PPDP para a anonimização de dados não estruturados incluem k-segurança, k-*confusability* e t-plausibilidade [33].

K-segurança é um método que considera que um documento é k-seguro se os atributos relacionados a uma dada entidade também podem ser associados a pelo menos k-1 outras entidades no documento. Contudo, esta técnica é considerada apenas praticável para domínios muito restritos.

K-*confusability* é o modelo em que um documento, com uma certa entidade sensível, é considerado k-*confusable* se, havendo um classificador que tem como *input* documentos, este revela como *output*, pelo menos, k outras entidades. Para tal funcionar, as entidades devem ser estáticas e os documentos devem coincidir com os corpora dados ao classificador para treino.

T-plausibilidade considera um documento t-plausível se, em relação a uma ontologia, pelo menos t documentos possam ser generalizados pelos seus termos. Este método depende do tamanho do documento, do número de entidades sensíveis e da granularidade de conhecimento.

2.2.8 Corpora para Avaliação da Anonimização

Muitos dos corpora utilizados para avaliar a anonimização em documentos de texto ou se focam no domínio clínico ou no processo de de-identificação, em vez de considerarem a anonimização. Esta conclusão foi referida por Pilán et. al [46], os quais acabaram por desenvolver um corpus que pretende evitar estas limitações. Este corpus, ao contrário de muitos outros, contém informação real de vários indivíduos reais, em documentos bem detalhados e que não tenham passado por qualquer processo de anonimização anterior. O corpus é baseado em casos de justiça do Tribunal Europeu dos Direitos Humanos (TEDH) e inclui apenas julgamentos publicados após 2018. No entanto, aborda apenas o idioma inglês, servindo apenas para uma possível futura avaliação da anonimização de dados não estruturados que se foque na língua inglesa, não sendo o caso deste projeto.

Capítulo 3

Seleção de Modelo para Deteção de Entidades

Neste capítulo é descrito o processo de seleção de modelos responsáveis pela identificação de termos a anonimizar num documento. Para tal, foram utilizados vários corpora, os quais tiveram de ser processados de diferentes formas para que pudessem ser treinados e avaliados pelas ferramentas usadas. Este processamento consiste em colocar todos os corpora no formato IOB e incluir as mesmas *labels*. Inicialmente, alguns corpora incluíam mais *labels* que outros, portanto foi decidido que todos os corpora apresentariam apenas as *labels* PER (pessoa), ORG (organização) e LOC (local), visto que estas são as classificações comuns a todos os corpora utilizados.

Apresenta-se em seguida o processo de seleção do modelo para a língua portuguesa.

3.1 Dados

Para a seleção de um modelo que atuasse sobre documentos em português, foram utilizados os 4 corpora apresentados na secção 2.2.6 que incluem a língua portuguesa:

- **LeNER-Br** [37], um corpus que se foca em documentos legais escritos em português do Brasil.
- **SECOND HAREM** [25] contém textos de blogs, Wikipédia, questões usadas para avaliação de QA, assim como texto de jornais.
- **MULTICONER** [39] possui texto de 3 domínios: frases da Wikipédia, questões QA e *queries* de pesquisa.
- **WIKIANN** [48] que inclui artigos da Wikipédia.

3.2 Pré-processamento de Dados

Os dados anteriormente apresentados foram então sujeitos a um pré-processamento, de forma a estarem aptos para serem utilizados na seleção do modelo:

- **LeNER-Br** [37] já se encontrava em formato IOB, portanto só foi necessário remover as *labels* JURISPRUDÊNCIA, LEGISLAÇÃO e TEMPO, assim como substituir as *labels* ORGANIZAÇÃO, PESSOA e LOCAL, por ORG, PER e LOC, respetivamente.
- **SECOND HAREM** [25] estava originalmente em XML, tendo de ser convertido para formato IOB (com extensão conll). Tal como no corpus anterior, foram mantidas apenas as *labels* PESSOA, LOCAL e ORGANIZAÇÃO, convertidas para PER, LOC e ORG, e foram removidas as restantes *labels*, as quais incluíam TEMPO, VALOR, OBRA, ABSTRACAO, COISA, EVENTO, OUTRO e ACONTECIMENTO.
- **MULTICONER** [39] foi processado para ficar de acordo com o formato IOB. No que diz respeito às *labels*, várias das suas *labels* poderiam ser incluídas dentro das categorias que foram consideradas. Para a categoria LOC foram inseridas as *labels* LOC, HumanSettlement, OtherLOC, Station e Facility; para a categoria ORG foram incluídas as *labels* MusicalGRP, SportsGRP, PrivateCorp, PublicCorp, Facility, CarManufacturer, AerospaceManufacturer e ORG; na categoria PER foram consideradas Scientist, OtherPER, Cleric, SportsManager, Politician, Athlete, Artist e PER.
- **WIKIANN** [48], inicialmente, estava em formato parquet, tendo sido depois convertido para conll (formato IOB). Não foram alteradas quaisquer *labels*, pois o corpus apenas considerava PER, ORG e LOC.

No fim do processamento dos corpora ainda foi criado um corpus extra, ao qual nomearemos de **ALLDATA**, o qual diz respeito à junção de todos os corpora anteriores.

3.3 Modelos de Deteção de Entidades

3.3.1 Treino de Modelos

Para o treino dos modelos, foram testadas diferentes ferramentas, as quais foram selecionadas com base no facto de suportarem a língua portuguesa e/ou serem de fácil utilização e aprendizagem. Estas ferramentas são:

- **Spacy**[1], tendo sido utilizados para esta os parâmetros por omissão. Spacy apresenta várias *pipelines* já treinadas para diversos idiomas. Estas *pipelines* também são treinadas para melhorar a eficiência na obtenção de resultados, ou para melhorar a sua exatidão. Para a seleção do modelo em português foi utilizada a *pipeline* pt_core_news_lg, a qual foi treinada para a língua portuguesa e para apresentar resultados com maior exatidão.
- **NLTK**[2], recorrendo à arquitetura CRF. Para os modelos treinados utilizando esta ferramenta, foram testados diferentes hiperparâmetros com o objetivo de ser selecionado o melhor modelo. Para cada um dos hiperparâmetros c1 e c2 foram testados 10 valores aleatórios

[1] `https://spacy.io/`
[2] `https://www.nltk.org/`

de uma distribuição exponencial com média de 0.5. O número máximo de iterações teve como valores testados todos aqueles de um intervalo entre 50 e 150, de 10 em 10.

- **Spacy + NLTK**, ou seja, foram comparadas as previsões dos modelos resultantes das ferramentas anteriores, dando prioridade às de NLTK, devido a resultados ligeiramente melhores, os quais serão apresentados mais adiante. Se a previsão de NLTK tivesse uma confiança maior ou igual a 0.60, então seria considerada, caso contrário, seria tida em conta a previsão do modelo treinado em Spacy.
- **Apache OpenNLP**[3] foi testada com variação de alguns hiperparâmetros. O hiperparâmetro *cutoff* variou entre os valores pares do intervalo de 0 a 20. O número de iterações variou entre 50 e 200, sempre com valores de 10 em 10.
- **Transformer** liaad/NER_harem_bert-large-portuguese-cased[4], esta é uma versão *fine-tuned* de BERTimbau Large [53] no corpus HAREM.

3.3.2 Modelos Externos

Para além das ferramentas anteriormente mencionadas, também foram utilizados vários modelos LLM de acesso aberto para observar o comportamento que estes apresentam para os mesmos corpora. Os modelos LLM utilizados foram os seguintes:

- **Together AI**[5], esta plataforma disponibiliza vários modelos que podem ser utilizados para a avaliação pretendida. Neste caso foi utilizado o modelo meta-llama/Llama-3.3-70B-Instruct-Turbo-Free, cuja utilização era grátis. Realça-se que a versão grátis da Together AI apenas fornecia 2$ de créditos, e um limite de 60 pedidos por minuto. Desta forma, foi apenas viável avaliar este modelo com os corpora de teste LENER-BR e HAREM, uma vez que são menores. Destaca-se ainda o facto dos resultados do modelo estarem também dependentes do *prompt* que lhe foi dado.
- **Gemini**[6] também disponibiliza vários modelos, ao quais se pode apresentar um *prompt* para avaliar o seu desempenho com os corpora NER. O modelo utilizado neste caso foi Gemini 1.5 Flash, o qual possui os limites de 1048576 *tokens* dados como *input* e 8192 *tokens* de *output*. Devido a estes limites os corpora utilizados também foram os mais pequenos, LENER-BR e HAREM. E o resultado de cada um teve de ser obtido dividindo o corpus em várias partes e realizando diversos pedidos, uma vez que não é possível fornecer o corpus inteiro ao modelo.
- **Chat-GPT**[7], foi apresentado um *prompt* com o objetivo do Chat poder apresentar um ficheiro json com os resultados encontrados. Foi utilizada a versão gratuita do Chat-GPT

[3] https://opennlp.apache.org/
[4] https://huggingface.co/liaad/NER_harem_bert-large-portuguese-cased
[5] https://www.together.ai/
[6] https://ai.google.dev/
[7] https://chatgpt.com/

(Chat-GPT 4.5), logo o fornecimento de ficheiros dos corpora é limitado por dia, e esta avaliação teve de ser realizada ao longo de vários dias por este motivo. Por esta razão, usou-se apenas um corpus, LeNER-BR, separado em várias partes, pois no caso de um ficheiro muito grande, o Chat-GPT não o processa inteiramente.

Os resultados destes modelos externos serviram apenas para uma questão de comparação. Independentemente de serem piores ou melhores que os restantes modelos, não poderão ser utilizados devido à sensibilidade dos dados que irão ser fornecidos ao modelo selecionado, os quais não deverão ser acedidos por um modelo externo.

3.4 Resultados

Para concretizar a avaliação dos vários modelos e compará-los entre si, foi necessário dividir cada corpus em três partições: partição de treino, partição de validação e partição de teste. Alguns dos corpora já vinham devidamente particionados, como LeNER-Br, MULTICONER e WIKIANN. No entanto, para o SECOND HAREM, que não vinha particionado, foram baralhados os 129 documentos que este possuía, e particionados em conjunto de treino com 90 documentos ($\approx 70\%$), conjunto de validação com 20 documentos ($\approx 15\%$) e conjunto de teste com 19 documentos ($\approx 15\%$). Quanto ao corpus ALLDATA, cada partição sua é o resultado da junção de todas as mesmas partições dos outros corpora, ou seja, a concatenação de todas as partições de treino deu origem à partição de treino de ALLDATA, e o mesmo acontece com as restantes.

Durante a obtenção dos resultados dos testes aos modelos foi possível observar três métricas:

- o recall que analisa a quantidade de termos bem identificados e classificados (TP) relativamente à totalidade de termos identificados, quer sejam classificados ou não (TP+FN), e é dada por:

$$recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.1}$$

- a precisão que consiste na quantidade de termos identificados e bem classificados (TP) face a quantidade de todos os termos identificados e classificados (TP + FP) e que é dada por:

$$precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.2}$$

- o *f1-score*, que considera ambas as métricas anteriores dado por :

$$f1score = 2 * \frac{\text{precisão} * \textit{recall}}{\text{precisão} + \textit{recall}} \tag{3.3}$$

Será dada mais atenção ao *f1-score*, uma vez que obtém um balanço entre as outras duas métricas, especialmente se o corpus for desequilibrado.

Para cada ferramenta utilizada, foram produzidos 5 modelos, um para cada corpus, treinados e validados com as devidas partições, sendo que a validação consiste na testagem de vários hiperparâmetros com fim a conseguir o modelo com os melhores hiperparâmetros. De seguida,

cada modelo produzido foi testado na partição de teste de cada corpus, de forma a verificar o seu comportamento com dados conhecidos. E por fim, de forma a comparar resultados, foram todos testados com a partição de teste de ALLDATA. Na tabela 3.1 são apresentados os resultados obtidos quanto às devidas partições de teste.

Como pode ser observado na tabela, os resultados da combinação Spacy+NLTK mostram-se instáveis, sendo o melhor resultado com o corpus Multiconer, mas os piores com os corpora ALLDATA e Wikiann.

Para além disso, é possível notar que os modelos externos não apresentaram melhores performances, ou foram equivalentes aos restantes como é, por exemplo, o caso de Together API com o corpus LeNER-BR ou tiveram resultados piores, como é o caso do Chat-GPT também com o corpus LeNER-BR. Os resultados destes modelos quanto aos corpora Multiconer, Wikiann, ALLDATA e SECOND HAREM (este apenas no caso do CHAT-GPT) não foram obtidos devido às limitações mencionadas na secção 3.3.2.

Foram escolhidos os modelos NLTK treinado com ALLDATA e o Transformer também treinado com ALLDATA, uma vez que são estes aqueles que apresentam os melhores resultados no cenário em que os vários corpora estão presentes. Ambos foram implementados, e no final um foi escolhido um com base nos resultados da avaliação das soluções finais.

Tabela 3.1: Resultados dos modelos produzidos relativamente à partição de teste do respetivo corpus.

Corpora	**Ferramenta**	**F1-Score**	**Recall**	**Precision**
LeNER-BR	Spacy	0.56	0.51	0.63
	NLTK	0.55	0.52	0.62
	Spacy + NLTK	0.47	0.45	0.80
	Apache OpenNLP	0.57	0.49	0.69
	Transformer	0.85	0.92	0.82
	TogetherAI	0.74	0.73	0.76
	Gemini API	0.59	0.57	0.63
	Chat-GPT	0.33	0.22	0.70
SECOND HAREM	Spacy	0.56	0.51	0.63
	NLTK	0.55	0.52	0.62
	Spacy + NLTK	0.47	0.45	0.80
	Apache OpenNLP	0.57	0.49	0.69
	Transformer	0.85	0.92	0.82
	TogetherAI	0.74	0.73	0.76
	Gemini API	0.59	0.57	0.63
MULTICONER	Spacy	0.70	0.64	0.78
	NLTK	0.75	0.71	0.81
	Spacy + NLTK	0.89	0.90	0.90
	Apache OpenNLP	0.62	0.52	0.78
	Transformer	0.74	0.70	0.80
WIKIANN	Spacy	0.84	0.83	0.85
	NLTK	0.84	0.85	0.84
	Spacy + NLTK	0.33	0.28	0.40
	Apache OpenNLP	0.84	0.83	0.86
	Transformer	0.71	0.76	0.70
ALLDATA	Spacy	0.70	0.68	0.73
	NLTK	0.74	0.69	0.81
	Spacy + NLTK	0.33	0.21	0.80
	Apache OpenNLP	0.40	0.32	0.55
	Transformer	0.74	0.71	0.82

3.5 Sumário

Após a consideração de todos os modelos criados, assim como os resultados que os mesmos geraram, concluiu-se que as melhores opções são aquelas que foram treinadas com o corpus ALL-DATA, uma vez que esta inclui todos os corpora considerados, possuindo assim uma maior cobertura de tópicos. Dentro dos modelos treinados com este corpus, foram selecionados o modelo da ferramenta NLTK com arquitetura CRF e o modelo com arquitetura Transformer, uma vez que estes apresentam os melhores resultados relativamente a este corpus. Para além disso, estes modelos também demonstram ser constantes nos resultados, em geral.

Capítulo 4

Sistema de Anonimização de Dados Não Estruturados

Este capítulo descreve o funcionamento do sistema, apresentado de forma geral na figura 4.1. Para além disso, também detalha a integração deste sistema com o projeto HYDE.

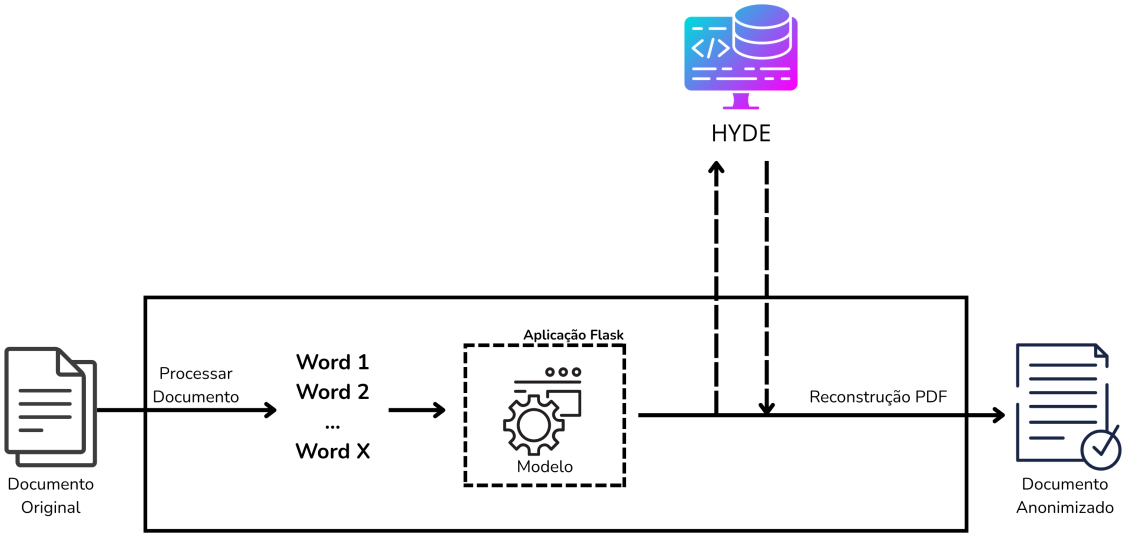

Figura 4.1: Esquema do fluxo geral do Sistema de Anonimização de Dados Não Estruturados

4.1 Processamento do Documento

O PDF original do documento a anonimizar é processado página a página. Dentro de cada uma das páginas, dá-se a separação de palavras, as quais são de seguida enviadas como corpo JSON de um pedido HTTP para a aplicação Flask que contém o modelo, dando início ao processo de identificação dos termos a anonimizar.

4.2 Identificação de Termos a Anonimizar

Após a receção da resposta do modelo (a qual vem em formato JSON), para cada página são guardados os respetivos termos que devem ser anonimizados. A figura 4.2 representa o esquema

relativo ao procedimento de identificação de termos. A criação das aplicações que contêm os modelos e que, dessa forma, são responsáveis pela identificação dos termos a anonimizar, será detalhada em seguida.

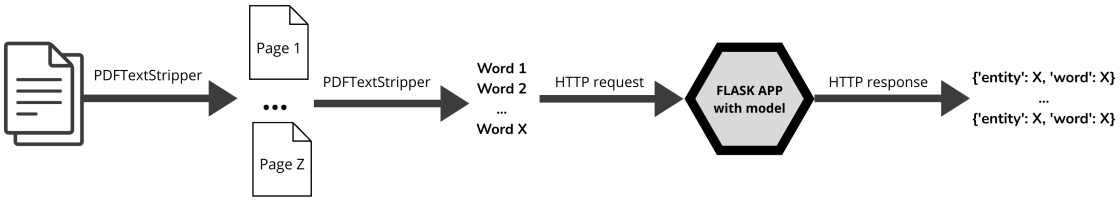

Figura 4.2: Esquema do processamento do PDF para identificação de termos

4.2.1 Modelos

Para possibilitar que uma outra aplicação possa chamar e utilizar o modelo selecionado, este foi integrado numa aplicação Python Flask. Esta aplicação inicia um servidor e define um *endpoint* /predict, o qual recebe dados de texto e os processa com o modelo, tendo como resultado as categorias identificadas no texto fornecido.

Para ambos os modelos selecionados anteriormente, foi adicionada uma camada extra para a identificação de termos como números de telefone portugueses, emails, códigos postais e números de identificação nacional através de padrões regex.

Foram desenvolvidas duas aplicações Python, uma para cada modelo, visto que os modelos apresentam diferentes arquiteturas, e como tal são necessárias formas diferentes de processar os dados para cada um.

Modelo Transformer

No caso do modelo que se trata de um Transformer, existe a restrição de não ser possível processar mais do que 512 *tokens* de uma vez. Sendo assim, o texto é particionado em blocos de 512 *tokens* ou menos, concatenando-se de seguida os resultados obtidos. A figura 4.3 representa o esquema de funcionamento da aplicação Flask do modelo Transformer.

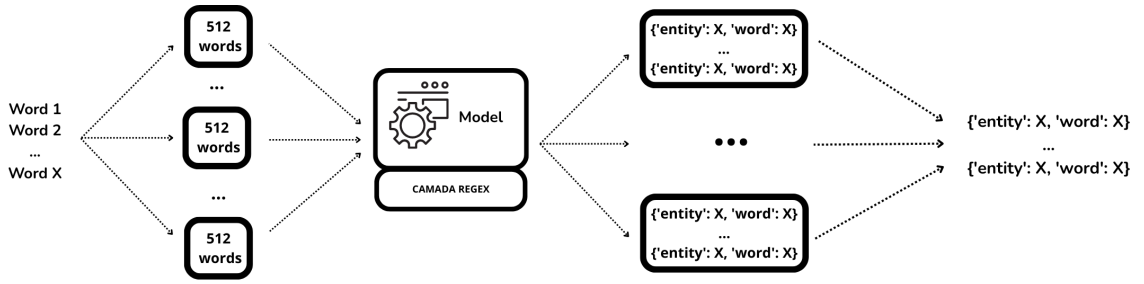

Figura 4.3: Esquema do Funcionamento do Modelo Transformer

Modelo NLTK

Para implementar a aplicação com o modelo NLTK, não é necessário dividir o texto em blocos, fornecendo-o inteiramente ao modelo. Como resultado, a resposta será uma lista de tuplos, cada qual conterá a palavra e a sua classificação, tal como é apresentada na figura 4.4.

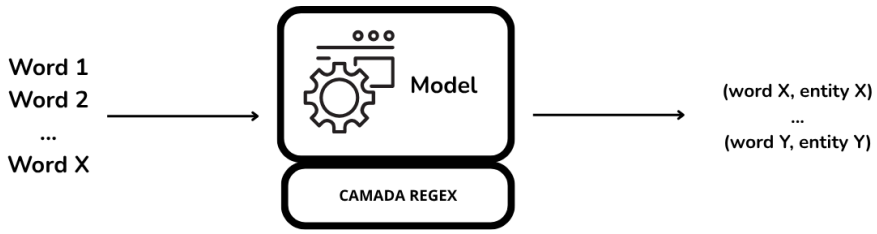

Figura 4.4: Esquema do Funcionamento do Modelo NLTK

4.3 Anonimização do Documento

Já sendo possível chamar o modelo, de forma a este poder ser usado para a identificação de termos sensíveis num documento PDF, passou-se à implementação da anonimização do documento. Nesta fase, utiliza-se a resposta do modelo para realizar a construção do PDF anonimizado. A aplicação resultante desta implementação será referida como PDFAnonymizer.

Para a realização do processamento e da produção de um PDF foi utilizada a biblioteca Apache PDFBox. Esta foi escolhida, pois é *open-source* e compatível com Java (linguagem utilizada para anonimizar o documento). Existiam outras bibliotecas como iText e PDFJet, mas a primeira exigia que o código fosse *open-source*, enquanto a segunda é comercial.

4.3.1 Construção do PDF

Após a obtenção de quais palavras devem ser anonimizadas em cada página, prossegue-se para a construção do PDF anonimizado, a qual é esquematizada na figura 4.5.

O objetivo deste processo é obter um PDF semelhante ao original, tentando manter ao máximo a sua formatação, no entanto, com os devidos termos anonimizados.

Esta tarefa foi realizada página a página, em que para cada página as devidas palavras eram obtidas, assim como as suas posições na página. Para representar cada palavra, foi implementada a classe WordWithLocation, a qual possui os seguintes campos:

- `word`, o conteúdo da palavra
- `initX`, o qual identifica a posição horizontal do início da palavra
- `initY`, o qual identifica a posição vertical do início da palavra
- `endX`, identifica a posição horizontal do final da palavra
- `endY`, identifica a posição vertical do final da palavra

- `width`, que corresponde à largura da palavra
- `height`, que corresponde à altura da palavra
- `fontSize`, o tamanho da *font* da palavra
- `font`, a *font* da palavra
- `isFootnote`, identifica se a palavra faz parte do rodapé de uma página
- `previousWord`, é um outro objeto WordWithLocation que corresponde à palavra anterior à atual, caso esta exista
- `inline`, valor booleano que diz se a palavra faz parte do parágrafo da palavra anterior.
- `spacingToNextPar`, representa o espaçamento que existe entre o parágrafo atual e o próximo
- `Region`, objeto que representa um parágrafo, incluindo os espaçamentos horizontais e verticais entre as palavras do mesmo.

As palavras detetadas para anonimização têm o seu valor de `word` substituído pelo respetivo valor que as oculta. Se as páginas tiverem imagens estas também são detetadas com as respetivas posições. Após isto, cada página é reconstruída e guardada com as palavras e as imagens anteriormente obtidas. No final, as páginas guardadas são concatenadas pela ordem correta num só ficheiro PDF, o resultado final.

A técnica de anonimização implementada neste trabalho foi a Supressão, e inicialmente, para facilitar este processo, as palavras anonimizadas eram substituídas por termos com o mesmo número de caracteres, por exemplo, se os termos a anonimizar fossem Raquel Domingos, estes seriam substituídos no documento final por ****** ********. Contudo, mais tarde, foi testado ocultar estes termos com palavras de mais ou menos caracteres. Para que este teste fosse bem-sucedido, foi necessário aperfeiçoar a localização de cada palavra obtida do PDF. Ou seja, as palavras podiam aparecer antes de onde originalmente estariam ou depois. No caso de haver mais caracteres, e as palavras terem de aparecer mais adiante da sua posição, foi necessário passar palavras de uma página para a seguinte.

4.3.2 Limitações

Apesar de ser possível recriar um PDF novo com os termos indicados anonimizados, o processo tem algumas limitações.

Uma destas limitações diz respeito aos tipos de letra (*fonts*) utilizados nos ficheiros PDF, pois existe uma infinidade destes que podem ser utilizados num documento, mas a aplicação criada não os consegue suportar todos. Na realidade, para facilitar o processo, o documento inicial utilizado para a reconstrução tinha a *font* Consola, pois esta apresenta o mesmo espaçamento entre palavras

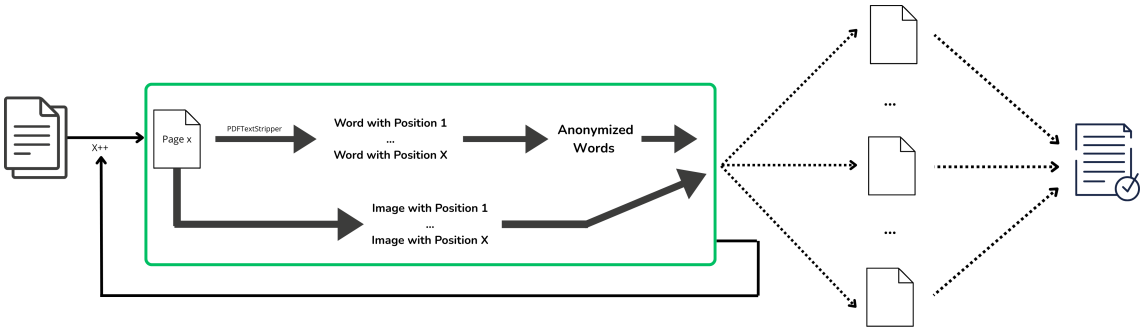

Figura 4.5: Esquema da construção do PDF final

e as letras também ocupam o mesmo espaço. Entretanto, foi possível o PDF original possuir outras *fonts*, mas o PDF final será sempre construído recorrendo à *font* Consola.

Outra limitação trata-se do número de colunas do documento. A aplicação apenas processa adequadamente documentos com uma coluna apenas, pois devido às diferenças que seriam necessárias realizar no processo de reconstrução do documento, seria necessária uma outra aplicação para reconstruir documentos com duas ou mais colunas.

Apesar do esforço colocado em melhorar ao máximo a reconstrução do PDF, conforme a diferença de formatos de PDFs dados como *input*, poderão surgir *edge-cases* que acabem por comprometer a formatação final, sendo este um processo que deve estar em contínuo desenvolvimento devido à sua complexidade.

4.4 Integração com o Projeto HYDE

Nesta secção é descrito o processo necessário para conseguir adicionar a funcionalidade da anonimização de documentos ao projeto HYDE.

A figura 4.6 mostra o estado final da infraestrutura após a integração ficar completa. O Traefik funciona como uma API *gateway* e um *loadbalancer* para o *backend* e *frontend*, assim como um *middleware* de autenticação (gerida pelo Keycloak) do *backend*.

O *backend* do projeto HYDE comunica com os microsserviços DMS (microsserviço de documentos) e NMS (microsserviço de notificações) e com a aplicação PDFAnonymizer, que também comunica com os mesmos microsserviços. O PDFAnonymizer serve como um interlocutor entre o modelo e o projeto HYDE, transmitindo os pedidos para o modelo e comunicando a resposta ao projeto HYDE. O PDFAnonymizer também é responsável pelos pedidos e realização da anonimização dos documentos, conforme os termos que o utilizador decide que devem ser anonimizados.

Os ficheiros são guardados num *bucket* Amazon S3, gerido pelo DMS. Todas as bases de dados utilizadas são bases de dados PostgresSQL numa instância de Amazon RDS, à exceção da base de dados do PDFAnonymizer, a qual é local ao *container* da aplicação.

De forma a automatizar o processo de entrega contínua do PDFAnonymizer e do modelo, foram criados *scripts* de automatização, como Dockerfiles e Jenkinsfile.

Este processo consistiu em três fases: a integração da aplicação Flask, a implementação de

pedidos HTTP na aplicação PDFAnonymizer e também a sua integração no ambiente de desenvolvimento, e por fim, a criação de pedidos HTTP no lado do projeto HYDE.

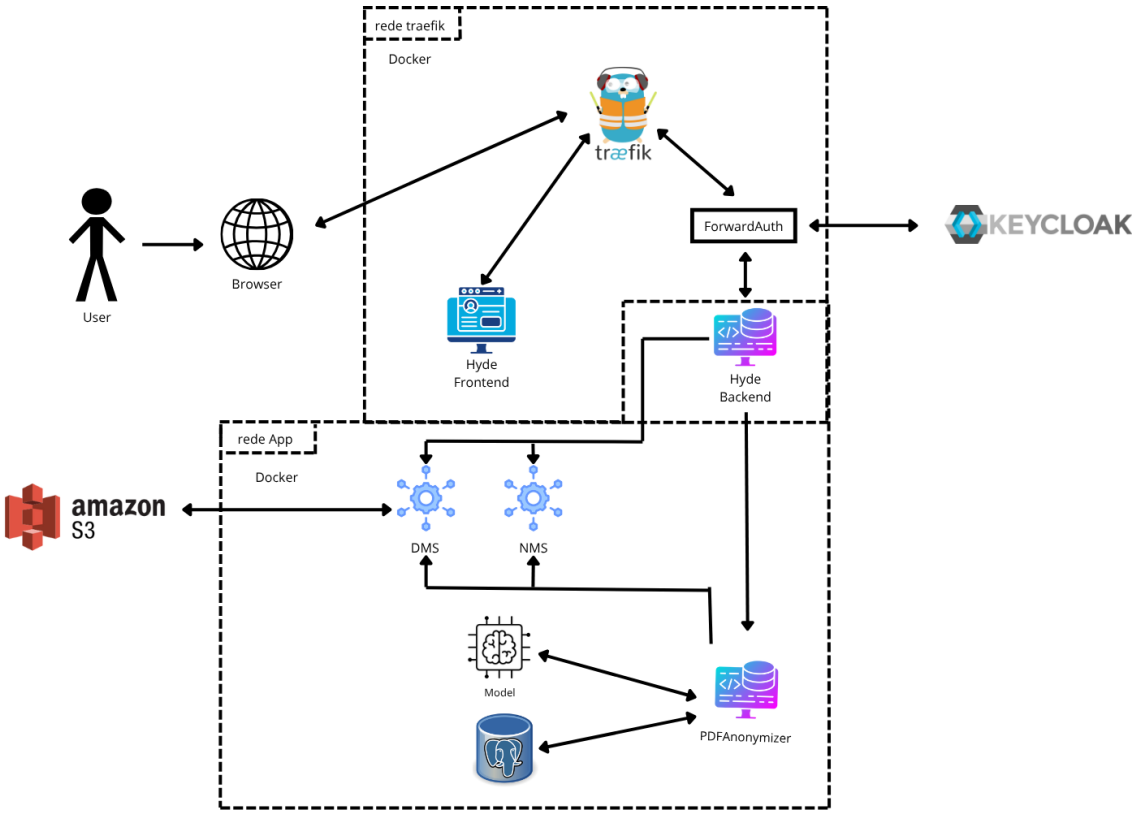

Figura 4.6: Esquema da infraestrutura do sistema

4.4.1 Integração do Modelo

Uma das necessidades do projeto é a colocação do modelo selecionado em ambiente de desenvolvimento para que possa ser acedido pela aplicação PDFAnonymizer e consequentemente pelo projeto HYDE.

O modelo selecionado a ser colocado em ambiente de desenvolvimento foi o modelo com a arquitetura Transformer (escolhido após a avaliação final detalhada no Capítulo 5). Para que esta colocação fosse possível, foi utilizado o Docker [9] que criou uma imagem que continha a aplicação, e esta foi enviada para o Nexus da empresa. O processo realizado está ilustrado em 4.7.

4.4.2 Integração da Aplicação PDFAnonymizer

Para a realização desta fase do processo, para além da implementação dos *endpoints* HTTP e da colocação da aplicação em ambiente de desenvolvimento, foi necessário incluir dois microsserviços da empresa: um microsserviço de notificações, que é responsável pela criação e armazenamento de notificações e um microsserviço de documentos, que faz o *upload* e *download* dos documentos, guardados em Amazon S3.

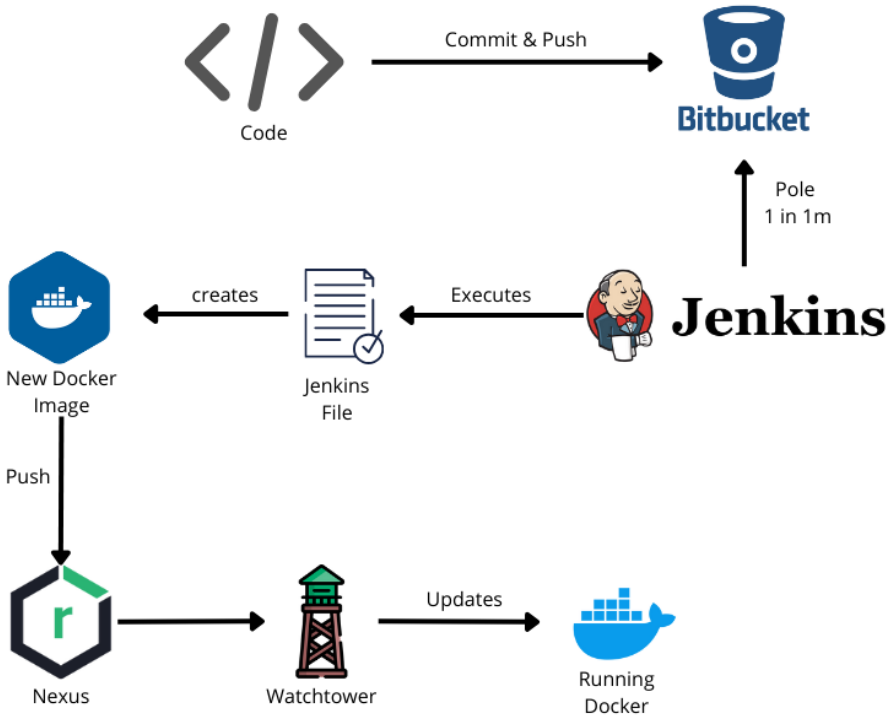

Figura 4.7: Esquema do processo de *deploy* para ambiente de desenvolvimento

Para além disso também foi criada uma base de dados, a qual contém apenas uma tabela, file_anonimization para guardar os resultados das anonimizações realizadas.

Foram adicionados dois *endpoints* HTTP à aplicação:

- `api/model`, *endpoint* POST que recebe no seu corpo um objeto que representa uma anonimização, FileAnonymizationAnonymizer. Este objeto tem de conter obrigatoriamente o id da anonimização no projeto HYDE e o UUID do documento a ser anonimizado. Este pedido irá então descarregar o documento com o devido UUID, a partir do microsserviço de documentos, e irá enviá-lo ao modelo, realizando o procedimento descrito na secção 4.2. Após isso guarda o resultado na base de dados, no formato observado na listagem 4.1 (transformado em String). Por último, envia uma notificação à aplicação do projeto HYDE, através do microsserviço das notificações, de forma a alertar que o processamento do documento pelo modelo já terminou. Por fim, é também enviado o resultado do modelo ao projeto HYDE.

- `api/anonymization`, *endpoint* POST que recebe um objeto anonymizationRequest, o qual deve conter obrigatoriamente, o id da anonimização no projeto HYDE e o payloadInfo desta anonimização, que no caso se trata de um objeto com formato igual ao da listagem 4.1. Este cria um novo PDF, semelhante ao original, mas com os devidos termos anonimizados. É retornado o PDF criado.

Listagem 4.1: Exemplo de resposta do modelo


```
{"schema":[
  {"name":"public","pages":[
    {"name":"page 1","terms":[
      {"entity":"B-ORGANIZACAO","originalValue":"Tribunal da Relação de
         Lisboa","toAnonymize":false,"selected":false},
      {"entity":"B-PESSOA","originalValue":"SARA REIS MARQUES","
         toAnonymize":false,"selected":false},
      {"entity":"B-ORGANIZACAO","originalValue":"SUPREMO TRIBUNAL DE
         JUSTIÇA","toAnonymize":false,"selected":false},
      {"entity":"B-ORGANIZACAO","originalValue":"Supremo Tribunal de
         Justiça","toAnonymize":false,"selected":false},
      {"entity":"B-ORGANIZACAO","originalValue":"STJ","toAnonymize":
         false,"selected":false},
      {"entity":"B-ORGANIZACAO","originalValue":"5ª Secção do Tribunal
         da Relação de Lisboa","toAnonymize":false,"selected":false}
    ]},
    {"name":"page 2","terms":[
      {"entity":"B-ORGANIZACAO","originalValue":"Tribunal da Relação","
         toAnonymize":false,"selected":false},
      {"entity":"B-ORGANIZACAO","originalValue":"Supremo Tribunal de
         Justiça","toAnonymize":false,"selected":false},
      {"entity":"B-ORGANIZACAO","originalValue":"Colendo Tribunal","
         toAnonymize":false,"selected":false},
      {"entity":"I-LOCAL","originalValue":"nº 3 do","toAnonymize":false
         ,"selected":false},
      {"entity":"B-ORGANIZACAO","originalValue":"Tribunal","toAnonymize
         ":false,"selected":false},
      {"entity":"B-ORGANIZACAO","originalValue":"STJ","toAnonymize":
         false,"selected":false}
    ]}
  ]}],
  "selectedCountry":null,
  "selectedLanguage":null
}
```


Após a implementação de todos estes *endpoints*, pode ser então realizada a integração da aplicação, que tal como a do modelo também pode ser observada em 4.7.

4.4.3 Implementação em HYDE

O projeto HYDE já tinha ligação com os microsserviços de notificações e de documentos. Foi-lhe adicionada a ligação com o PDFAnonymizer. Ressalva-se que o PDFAnonymizer em ambiente de desenvolvimento não pode ser acedido de outra forma se não através do projeto HYDE.

Como os *endpoints* de submeter um documento e guardá-lo em Amazon S3 já estavam feitos, uma vez que eram utilizados para a anonimização de dados estruturados, só foi necessário acrescentar um novo tipo de documento, para que fossem aceites PDFs.

Tendo sido preciso adicionar os seguintes pedidos:

- `api/document`, *endpoint* GET paginado que devolve todas as anonimizações de documentos realizadas.

- `api/document`, *endpoint* POST que recebe um objeto com os seguintes campos: título da anonimização, o país da anonimização, a língua da anonimização e ainda um outro objeto que contém o código, o UUID e o tipo do documento. Este pedido irá então inicializar uma anonimização nova, chamando assincronamente o pedido api/model da API do PDFAnonymizer. A resposta é um objeto FileAnonymization, o qual é apresentado na listagem 4.2. Assim que a resposta do modelo é recebida no método assíncrono, a anonimização é atualizada na BD, sendo que o estado passa a STARTED e o campo payloadInfo passa a ter o resultado do modelo.

- `api/document/model/{fileAnonymizationId}`, *endpoint* GET que recebe no seu *path* o id da anonimização e retorna ao utilizador o objeto 4.2.

- `api/document/{fileAnonymizationId}/submit`, *endpoint* PUT que recebe no seu *path* o id da anonimização e um objeto com o payLoadInfo, ou seja, um objeto como 4.1, mas com as alterações que o utilizador decidir fazer, que de momento é apenas decidir se o termo deve ser anonimizado ou não (através do campo selected do objeto JSON). Durante este pedido, o estado da anonimização é atualizado para SUBMITTED e é chamada a API do PDFAnonymizer, mais especificamente o pedido /api/anonymization. Este pedido irá devolver o PDF anonimizado, o qual será guardado em Amazon S3, assim como as suas informações que serão guardadas na base de dados.

- `api/document/{entityType}/download/{fileAnonymizationId}`, *endpoint* GET que recebe no seu *path* o tipo de documento (para o caso de PDFs é sempre FILE), e o id da anonimização a que este corresponde. Este pedido tem como objetivo a obtenção de informações sobre o documento a ser descarregado, o qual pode ser o PDF original, ou o anonimizado. Para tal, ainda recebe um parâmetro booleano isOriginal, o qual se for verdadeiro indica a intenção de descarregar o PDF original e se for falso trata-se então do PDF anonimizado. O objeto retornado com as informações do documento pode ser observado no exemplo da listagem 4.3, sendo que o entityId corresponde ao id da anonimização. Estas informações serão depois utilizadas pelo *frontend* da aplicação para fazer o *download* do ficheiro diretamente da Amazon S3.

Listagem 4.2: Exemplo de resposta obtida no pedido POST /api/document


```
{
  "error": null,
  "fileAnonimization": {
    "id": 3503,
    "anonymizationTitle": "Anonimizacao_Teste",
    "anonymizationCounter": null,
    "name": null,
    "fileOriginal": "748e4a86-2af0-4951-a22e-fed6ba427fa7",
    "status": "INITIALISING",
    "errorType": null,
    "fileOur": null,
```

```
    "payloadData": null,
    "payloadInfo": null,
    "output": null,
    "submittedDate": null,
    "completedDate": null,
    "closedDate": null,
    "statusDate": "2025-07-09T15:52:17.003451Z",
    "version": 0,
    "createdBy": "12cf0823-b59d-40d5-88f9-d31a90cd7f39",
    "createdDate": "2025-07-09T15:52:17.004754Z",
    "lastModifiedBy": "12cf0823-b59d-40d5-88f9-d31a90cd7f39",
    "lastModifiedDate": "2025-07-09T15:52:17.004754Z",
    "documentIn": {
      "id": 3554,
      "name": null,
      "input": true,
      "type": "inputDocumentFile",
      "entityType": "FILE",
      "entityId": 3503,
      "uuid": "748e4a86-2af0-4951-a22e-fed6ba427fa7",
      "originalName": "teste_pdf.pdf",
      "description": null,
      "createdBy": "12cf0823-b59d-40d5-88f9-d31a90cd7f39",
      "createdDate": "2025-07-09T15:52:17.312012Z",
      "lastModifiedDate": "2025-07-09T15:52:17.312012Z",
      "fileType": null
    }
  }
}
```


Listagem 4.3: Exemplo de resposta obtida com as informações sobre documento a descarregar


```
{
  "error": null,
  "document": [
    {
      "id": 3252,
      "name": null,
      "input": false,
      "type": "DIRECTAP",
      "entityType": "FILE",
      "entityId": 3201,
      "uuid": "4cd41439-d419-42de-8424-6cf36ce466d7",
      "originalName": "4cd41439-d419-42de-8424-6cf36ce466d7.pdf",
      "description": null,
      "createdBy": "12cf0823-b59d-40d5-88f9-d31a90cd7f39",
      "createdDate": "2025-07-08T10:55:39.295983Z",
      "lastModifiedDate": "2025-07-08T10:55:39.295983Z",
      "fileType": null
    }
  ]
}
```

Capítulo 5

Avaliação

5.1 Metodologia de Avaliação

A avaliação do sistema de anonimização de dados não estruturados teve como objetivo determinar a prestação de cada um dos dois melhores modelos selecionados (modelo Transformer e modelo NLTK com arquitetura CRF) e escolher o melhor entre eles.

Para a realização desta avaliação foi utilizado um corpus de currículos em português fornecidos pela Trust Systems. Nenhum dos dois modelos foi treinado, validado ou testado com dados deste corpus, tratando-se de dados completamente desconhecidos para estes. Ao todo foram utilizados 22 currículos. O motivo do tamanho reduzido do corpus deve-se à dificuldade de conseguir encontrar currículos em português, uma vez que a maior parte das pessoas envia currículos em inglês.

O processo de avaliação começou pela anotação dos currículos por parte de onze elementos da empresa (tendo cada elemento anotado dois currículos). Salienta-se que a anotação do documento não foi realizada pela autora da tese, para não haver qualquer enviesamento dos resultados. Os elementos responsáveis pela anotação tratam-se de desenvolvedores de software assim como gestores de projetos e a CEO da empresa.

De seguida, foram dados os mesmos currículos a um *Data Protection Officer* (DPO), de forma a que pudessem ser obtidas anotações a partir de uma perspetiva com conhecimento profissional relativamente à área de proteção de dados.

Devido às *labels* que, de momento, os modelos apenas conseguem identificar, foram descartados das anotações termos como datas, URLs, estado civil, idade, atividades ou cargos, telefones que não sejam portugueses e outros códigos ou números que não sejam o número de identificação nacional.

Após a obtenção destes dois grupos de currículos anotados (um grupo pelos elementos da empresa e outro grupo pelo DPO), os ficheiros originais foram fornecidos a cada um dos dois modelos anteriormente mencionados, com a finalidade de entender qual dos dois identificava dados mais relevantes para a anonimização.

Para analisar os resultados dos modelos e poder compará-los, tanto entre si, como com as anotações obtidas, foram selecionadas as métricas *f1-score*, precisão e *recall* ao nível dos *tokens*. Cada

token que foi anotado e que foi corretamente identificado e classificado pelo modelo será considerado um *True Positive* (TP). Caso um *token* não tenha sido anotado e mesmo assim tenha sido identificado pelo modelo, então trata-se de um *False Positive* (FP). Por último, quando um *token* foi anotado, mas não foi identificado, ou se foi identificado foi mal classificado, este considera-se um *False Negative* (FN). As fórmulas que dizem respeito a estas métricas apresentam-se de seguida:

$$\text{precisão} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.1}$$

$$\textit{recall} = \frac{TP}{\text{TP} + \text{FN}} \tag{5.2}$$

$$\textit{f1-score} = 2 * \frac{\text{precisão} * \textit{recall}}{\text{precisão} + \textit{recall}} \tag{5.3}$$

5.2 Resultados e Discussão

Na tabela 5.1, é possível verificar os resultados de ambos os modelos relativamente às anotações do DPO. Na tabela 5.2, são apresentados os resultados dos modelos comparando com as anotações realizadas pelos elementos da Trust Systems.

Ao observar ambas as tabelas, podemos perceber que o modelo NLTK possui resultados muito piores que o modelo Transformer. O modelo NLTK apresenta precisões muito baixas, indicando que muitos *tokens* irrelevantes são identificados, gerando muitos falsos positivos.

O modelo Transformer apresenta resultados satisfatórios, com algumas performances piores para certos currículos, como é o caso do CV12. No entanto, também podemos verificar que, dependendo de quem fez as anotações, os resultados vão variando, indicando a diferença de perceção do que deve ou não ser anonimizado por parte dos indivíduos.

Na tabela 5.3 observa-se as médias e os desvios-padrão obtidos para cada uma das tabelas de resultados. Destaca-se que as médias calculadas para o modelo Transformer são sempre superiores às do modelo NLTK.

Em geral, podemos concluir que o modelo Transformer apresenta os melhores resultados, sendo o modelo selecionado para ser integrado no projeto HYDE.

Tabela 5.1: Resultados dos modelos selecionados relativamente às anotações realizadas pelo DPO

Curriculo	Modelo	F1-Score	Recall	Precision
CV 1	Transformer	0.60	0.82	0.47
	NLTK	0.17	0.71	0.10
CV 2	Transformer	0.86	0.90	0.82
	NLTK	0.35	0.81	0.22
CV 3	Transformer	0.76	0.81	0.71
	NLTK	0.30	0.67	0.19
CV 4	Transformer	0.75	0.84	0.68
	NLTK	0.11	0.52	0.06
CV 5	Transformer	0.63	0.86	0.50
	NLTK	0.17	0.38	0.11
CV 6	Transformer	0.53	0.95	0.37
	NLTK	0.21	0.58	0.13
CV 7	Transformer	0.62	0.69	0.56
	NLTK	0.26	0.69	0.16
CV 8	Transformer	0.69	0.89	0.57
	NLTK	0.19	0.61	0.11
CV 9	Transformer	0.82	0.83	0.81
	NLTK	0.46	0.77	0.33
CV 10	Transformer	0.60	0.98	0.44
	NLTK	0.30	0.89	0.18
CV 11	Transformer	0.56	0.57	0.55
	NLTK	0.18	0.76	0.10
CV 12	Transformer	0.23	0.86	0.16
	NLTK	0.13	0.83	0.07
CV 13	Transformer	0.65	0.89	0.51
	NLTK	0.21	0.39	0.14
CV 14	Transformer	0.47	0.68	0.39
	NLTK	0.11	0.56	0.06
CV 15	Transformer	0.32	0.80	0.20
	NLTK	0.02	0.80	0.01
CV 16	Transformer	0.55	0.75	0.43
	NLTK	0.13	0.68	0.07
CV 17	Transformer	0.51	0.76	0.38
	NLTK	0.05	0.24	0.03
CV 18	Transformer	0.86	0.92	0.81
	NLTK	0.33	0.54	0.24
CV 19	Transformer	0.86	0.84	0.89
	NLTK	0.26	0.65	0.16
CV 20	Transformer	0.85	0.79	0.92
	NLTK	0.26	0.52	0.17
CV 21	Transformer	0.83	0.75	0.92
	NLTK	0.09	0.40	0.05
CV 22	Transformer	0.74	0.91	0.63
	NLTK	0.24	0.51	0.16

Tabela 5.2: Resultados dos modelos selecionados relativamente às anotações realizadas pelos elementos da empresa

Curriculo	Modelo	F1-Score	Recall	Precision
CV 1	Transformer	0.56	0.81	0.43
	NLTK	0.11	0.50	0.06
CV 2	Transformer	0.52	0.81	0.38
	NLTK	0.23	0.88	0.13
CV 3	Transformer	0.72	0.69	0.75
	NLTK	0.38	0.73	0.26
CV 4	Transformer	0.81	0.79	0.83
	NLTK	0.14	0.54	0.08
CV 5	Transformer	0.84	0.88	0.81
	NLTK	0.32	0.46	0.24
CV 6	Transformer	0.78	0.93	0.68
	NLTK	0.38	0.62	0.28
CV 7	Transformer	0.52	0.58	0.47
	NLTK	0.22	0.61	0.13
CV 8	Transformer	0.68	0.84	0.57
	NLTK	0.23	0.67	0.14
CV 9	Transformer	0.70	0.90	0.58
	NLTK	0.29	0.68	0.19
CV 10	Transformer	0.25	0.79	0.15
	NLTK	0.12	0.74	0.06
CV 11	Transformer	0.38	0.67	0.26
	NLTK	0.05	0.44	0.03
CV 12	Transformer	0.23	0.86	0.16
	NLTK	0.13	0.83	0.07
CV 13	Transformer	0.58	0.91	0.43
	NLTK	0.22	0.48	0.14
CV 14	Transformer	0.78	0.84	0.72
	NLTK	0.07	0.32	0.04
CV 15	Transformer	0.31	0.67	0.20
	NLTK	0.02	0.57	0.01
CV 16	Transformer	0.26	0.73	0.16
	NLTK	0.06	0.83	0.03
CV 17	Transformer	0.45	0.79	0.31
	NLTK	0.03	0.14	0.02
CV 18	Transformer	0.84	0.94	0.77
	NLTK	0.36	0.60	0.26
CV 19	Transformer	0.90	0.82	1.00
	NLTK	0.27	0.64	0.17
CV 20	Transformer	0.50	0.67	0.40
	NLTK	0.17	0.60	0.10
CV 21	Transformer	0.71	0.57	0.92
	NLTK	0.13	0.43	0.08
CV 22	Transformer	0.60	0.89	0.45
	NLTK	0.20	0.61	0.12

Tabela 5.3: Médias e Desvios-padrão dos resultados

Resultados	**Modelo**	**F1-Score**		**Recall**		**Precision**	
		Média	Desvio-Padrão	Média	Desvio-Padrão	Média	Desvio-Padrão
relativos às anotações do DPO	Transformer	0.65	0.17	0.82	0.09	0.58	0.22
	NLTK	0.21	0.10	0.61	0.16	0.13	0.07
relativos às anotações dos elementos da empresa	Transformer	0.59	0.20	0.79	0.11	0.52	0.25
	NLTK	0.19	0.11	0.59	0.17	0.12	0.08

Capítulo 6

Conclusão

Este projeto foca-se na extensão do projeto HYDE, projeto da Trust Systems que apenas anonimizava bases de dados. Pretendeu-se adicionar a funcionalidade de anonimizar dados não estruturados, mais especificamente ficheiros PDF.

A adição desta funcionalidade contou com duas fases: a primeira em que se treinou vários modelos NER e se escolheu o melhor entre eles e a segunda onde se criou um sistema que integra o melhor modelo selecionado com o projeto HYDE e que constrói o PDF final anonimizado.

Durante a fase de seleção dos modelos foram utilizadas várias ferramentas e corpora com o objetivo de se treinar modelos mais capazes. Dois foram selecionados, o modelo Transformer e o modelo NLTK com arquitetura CRF. No final após uma avaliação com um corpus composto por currículos fornecidos pela Trust Systems, o Transformer foi escolhido, uma vez que era aquele que identificava informação com mais precisão, e que apresentava em mais de metade dos currículos um *f1-score* igual ou superior a 0.60.

Na fase de implementação do sistema responsável pela integração do modelo e pela construção do PDF anonimizado, foi criada uma aplicação em Java, o PDFAnonymizer, recorrendo à biblioteca PDFBox para a construção do documento. Este sistema também permite a comunicação entre o modelo e o projeto HYDE.

6.1 Trabalho Futuro

Apesar dos resultados obtidos durante a realização do projeto, ainda existem muitos pontos a serem melhorados no sistema, os quais não foram possíveis de realizar por falta de tempo e limitações de tecnologias. Estas melhorias incluem:

- Novas ferramentas e corpora podem ser explorados com vista a criar um modelo melhor que o selecionado. Para além disso, o modelo atual, com exceção do regex, apenas identifica as *labels* que se referem a pessoas, organizações e locais. No futuro poderá ser criado um modelo que seja capaz de identificar mais *labels*.

- O projeto HYDE aborda diversos idiomas, para além do português, relativamente à anonimização de dados estruturados. Podem ser criados modelos que permitam a anonimização

de dados não estruturados para todos estes idiomas.

- O projeto HYDE implementa várias técnicas de anonimização, para além da supressão, para a anonimização de dados estruturados. Estas técnicas podem ser implementadas relativamente à anonimização de dados não estruturados.

- A construção do PDF final pode ser melhorada e é um processo que deve ir evoluindo, pois com a apresentação de novos tipos de formatos de PDFs, novas necessidades de formatação devem ser implementadas. A ferramenta utilizada foi PDFBox, uma ferramenta *open-source* que cumpria os requisitos do projeto, no entanto, outras ferramentas mais complexas e que não puderam ser utilizadas durante a tese podem ser testadas.

- O modelo escolhido foi treinado para ser capaz de identificar entidades num PDF com temas mais gerais, uma vez que não era necessário focar em nenhum tema (por exemplo, documentos legais, documentos médicos) em específico. No futuro, podem ser treinados e selecionados diferentes modelos, cada um para um tema diferente. Desta forma, o utilizador poderá escolher o tema que o seu PDF aborda na interface do projeto HYDE, e o devido modelo será utilizado para tal tema, com a finalidade de obter uma melhor performance.

- Novos tipos de dados não estruturados podem ser anonimizados, como imagens, assinaturas ou aúdio.

Bibliografia

[1] Ancora site. https://clic.ub.edu/corpus/en/ancora. [Online; Accessed: 2024-12-16].

[2] Tac 2017 site. https://tac.nist.gov/2017/index.html. [Online; Accessed: 2024-12-16].

[3] Angular site. https://angular.dev/, 2024. [Online; Accessed: 2024-12-03].

[4] Anonimatron site. https://realrolfje.github.io/anonimatron/, 2024. [Online; Accessed: 2024-11-29].

[5] Aws. https://aws.amazon.com/pt/rds/, 2024. [Online; Accessed: 2025-07-27].

[6] Spring site. https://spring.io/, 2024. [Online; Accessed: 2024-11-29].

[7] Trust systems site. https://www.trustsystems.eu/, 2024. [Online; Accessed: 2024-12-02].

[8] Amazon s3 site. https://aws.amazon.com/s3/?nc=sn&loc=0, 2025. [Online; Accessed: 2025-07-27].

[9] docker. https://www.docker.com/, 2025. [Online; Accessed: 2025-07-27].

[10] keycloak. https://www.keycloak.org/, 2025. [Online; Accessed: 2025-07-27].

[11] Postgressql site. https://www.postgresql.org/, 2025. [Online; Accessed: 2025-07-27].

[12] traefiklabs. https://traefik.io/traefik, 2025. [Online; Accessed: 2025-07-27].

[13] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and et al. Qwen technical report, 2023.

[14] Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Padó. Germeval 2014 named entity recognition shared task. 2014.

[15] William J Black, Fabio Rinaldi, and David Mowatt. Facile: Description of the ne system used for muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, 1998.

[16] Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. *The Groningen Meaning Bank*, pages 463–496. June 2017.

[17] Felix Böhlin. Detection anonymization of sensitive information in text: Ai-driven solution for anonymization, 2024. Linnaeus University, Faculty of Technology, Department of computer science and media technology (CM).

[18] Zhou Chaochao and Yang Bo. Text2struct: A machine learning pipeline for mining structured data from text, 2022.

[19] Gergely Márk Csányi, Dániel Nagy, Renátó Vági, János Pál Vadász, and Tamás Orosz. Challenges and open problems of legal document anonymization. *Symmetry*, 13(8), 2021.

[20] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin, editors, *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[22] Francisco Eduardo do Couto Soares Ramos. Anonimização automática de dados estruturados. Master's thesis, Faculdade de Ciências, Universidade de Lisboa, 2022.

[23] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10, 2014.

[24] Luís Bernardo Crisóstomo e Silva Rodrigues Esteves dos Santos. Anonimização automática. Master's thesis, Faculdade de Ciências, Universidade de Lisboa, 2024.

[25] Cláudia Freitas, Cristina Mota, Diana Santos, Hugo Gonçalo Oliveira, and Paula Carvalho. Second harem: Advancing the state of the art of named entity recognition in portuguese. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[26] James Gardner and Li Xiong. An integrated framework for de-identifying unstructured medical data. *Data Knowledge Engineering*, 68(12):1441–1451, 2009.

[27] Abbas Ghaddar and Phillippe Langlais. Wikicoref: An english coreference-annotated corpus of wikipedia articles. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan

Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 136–142, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[28] Jerry R. Hobbs, Douglas Appelt, John Bear, David Israel, Megumi Kameyalna, and Mabry Tyson. Fastus: A system for extracting information from text. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*, 1993.

[29] K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. University of sheffield: Description of the lasie-ii system as used for muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, 1998.

[30] Basra Jehangir, Saravanan Radhakrishnan, and Rahul Agarwal. A survey on named entity recognition — datasets, tools, and methodologies. *Natural Language Processing Journal*, 3:100017, June 2023.

[31] Pawel Jurczyk, James J. Lu, Li Xiong, Janet D. Cragan, and Adolfo Correa. Fril: A tool for comparative record linkage. *American Medical Informatics Associations (AMIA) 2008 Annual Symposium*, pages 440–444, 2008.

[32] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19:i180–i182, July 2003.

[33] Pierre Lison, Ildikó Pilán, David Sánchez, Montserrat Batet, and Lilja Øvrelid. Anonymisation models for text data: State of the art, challenges and future directions. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4188–4203. Association for Computational Linguistics, August 2021.

[34] Pan Liu, Yanming Guo, Fenglei Wang, and Guohui Li. Chinese named entity recognition: The state of the art. *Neurocomputing*, 473:37–53, 2022.

[35] Xiaohua Liu and Ming Zhou. Two-stage ner for tweets with clustering. *Inf. Process. Manage.*, 49(1):264–273, January 2013.

[36] Cedric Lothritz, Bertrand Lebichot, Kevin Allix, Lisa Veiber, Tegawendé F. Bissyandé, Jacques Klein, Andrey Boytsov, Anne Goujon, and Clément LeFebvre. Luxembert: Simple and practical data augmentation in language model pre-training for luxembourgish. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, Marseille, France, 2022. European Language Resources Association.

[37] Pedro Henrique Luz de Araujo, Teofilo E. de Campos, Renato R. R. de Oliveira, Matheus Stauffer, Samuel Couto, and Paulo Bermejo. Lener-br: A dataset for named entity recognition in brazilian legal text. In Aline Villavicencio, Viviane Moreira, Alberto Abad, Helena Caseli, Pablo Gamallo, Carlos Ramisch, Hugo Gonçalo Oliveira, and Gustavo Henrique Paetzold, editors, *Computational Processing of the Portuguese Language*, pages 313–323. Springer International Publishing, August 2018.

[38] Abdul Majeed and Sungchang Lee. Anonymization techniques for privacy preserving data publishing: A comprehensive survey. *IEEE Access*, 9:8512–8545, 2021.

[39] Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. Multiconer: A large-scale multilingual dataset for complex named entity recognition. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3798–3809, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[40] Meta. The llama 3 herd of models, 2024. Accessed: 2025-08-02.

[41] Suntherasvaran Murthy, Asmidar Abu Bakar, Fiza Abdul Rahim, and Ramona Ramli. A comparative study of data anonymization techniques. In *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pages 306–309. IEEE, 2019.

[42] Yichen Ning, Na Wang, Aodi Liu, and Xuehui du. Deep learning based privacy information identification approach for unstructured text. *Journal of Physics: Conference Series*, 1848, April 2021.

[43] Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175, 2013.

[44] OpenAI. Introducing gpt-5, 2025. Accessed: 2025-09-25.

[45] Nita Patil, Ajay Patil, and B.V. Pawar. Named entity recognition using conditional random fields. *Procedia Computer Science*, 167:1181–1188, 2020.

[46] Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. The text anonymization benchmark (tab): A dedicated corpus and evaluation framework for text anonymization. *Computational Linguistics*, 48(4):1053–1101, 2022.

[47] Irene Pérez-Diez, Raúl Pérez-Moragal, Adolfo López-Cerdán, Jose-Maria Salinas-Serrano, and María de la Iglesia-Vayál. De-identifying spanish medical texts - named entity recognition applied to radiology reports. *Journal of Biomedical Semantics*, 12(6), 2021.

[48] Afshin Rahimi, Yuan Li, and Trevor Cohn. Massively multilingual transfer for ner. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy, July 2019. Association for Computational Linguistics.

[49] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998.

[50] Diana Santos, Nuno Seco, Nuno Cardoso, and Rui Vilela. Harem: An advanced ner evaluation contest for portuguese. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odijk, and Daniel Tapias, editors, *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA).

[51] Xavier Schmitt, Sylvain Kubler, Jérémy Robert, Mike Papadakis, and Yves LeTraon. A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 338–343. IEEE, 2019.

[52] Thoudam Doren Singh, Kishorjit Nongmeikapam, Asif Ekbal, and Sivaji Bandyopadhyay. Named entity recognition for manipuri using support vector machine. In Olivia Kwong, editor, *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 2*, pages 811–818, Hong Kong, December 2009. City University of Hong Kong.

[53] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. *BERTimbau: Pretrained BERT Models for Brazilian Portuguese*, pages 403–417. 10 2020.

[54] Roomani Srivastava, Suraj Prasad, Lipika Bhat, Sarvesh Deshpande, Barnali Das, and Kshitij Jadhav. Medpromptextract (medical data extraction tool): Anonymization and hi-fidelity automated data extraction using nlp and prompt engineering, 2024.

[55] Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. Large language models are advanced anonymizers, 2024.

[56] Erik F. Tjong Kim Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

[57] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[59] Xinyu Wang, Yongliang Shen, Jiong Cai, Tao Wang, Xiaobin Wang, Pengjun Xie, Fei Huang, Weiming Lu, Yueting Zhuang, Kewei Tu, Wei Lu, and Yong Jiang. Damo-nlp at semeval-2022 task 11: A knowledge-based system for multilingual named entity recognition. In Guy Emerson, Natalie Schluter, Gabriel Stanovsky, Ritesh Kumar, Alexis Palmer, Nathan Schneider, Siddharth Singh, and Shyam Ratan, editors, *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1457–1468, Seattle, United States, July 2022. Association for Computational Linguistic.

[60] Yu Wang, Hanghang Tong, Ziye Zhu, and Li Yun. Nested named entity recognition: A survey. *ACM Transactions on Knowledge Discovery from Data*, 16(6), 2022.

[61] Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn Mattingly, Jiao Li, Thomas Wiegers, and Zhiyong lu. Assessing the state of the art in biomedical relation extraction: Overview of the biocreative v chemical-disease relation (cdr) task. *Database : the journal of biological databases and curation*, 2016, March 2016.

[62] Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. *OntoNotes: A Large Training Corpus for Enhanced Processing*. 2011.

[63] Emily M Weitzenboeck, Pierre Lison, Malgorzata Cyndecka, and Malcolm Langford. The gdpr and unstructured data: is anonymization possible? *International Data Privacy Law*, 12(3):184–206, August 2022.

[64] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2):100211, 2024.

[65] Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. Hyena: Hierarchical type classification for entity names. In Martin Kay and Christian Boitet, editors, *Proceedings of COLING 2012: Posters*, pages 1361–1370, Mumbai, India, December 2012. The COLING 2012 Organizing Committee.

[66] Shaodian Zhang and Noémie Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of Biomedical Informatics*, 46(6):1088–1098, 2013.

[67] GuoDong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 473–480, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

Apêndice A

Prompts utilizados na utilização de modelos externos

```
Você é um modelo de linguagem treinado para extrair entidades nomeadas de textos em português. Abaixo está um exemplo do que você deve fazer.
Extraí apenas as entidades ORGANIZACAO, PESSOA e LOCAL.

Texto:
\"\"\"Maria trabalha no Google em São Paulo desde 2022.\"\"\"

Resposta:
[
  {{ "entidade": "Maria", "tipo": "PESSOA" }},
  {{ "entidade": "Google", "tipo": "ORGANIZACAO" }},
  {{ "entidade": "São Paulo", "tipo": "LOCAL" }},
  {{ "entidade": "2022", "tipo": "DATA" }}
]

Agora, extraia as entidades do seguinte texto:

Texto:
\"\"\"{text}\"\"\"

Resposta:
"""
```

Figura A.1: Prompt utilizado com os modelos remotos da plataforma Together AI e de Gemini

```
Você é um modelo de linguagem treinado para extrair entidades
nomeadas de textos em português. Com o ficheiro dado txt dado,
processe e classifique as entidades com as seguintes labels:
Organizacao, Pessoa, Local. Caso sejam encontradas palavras
repetidas, devem ser apresentadas todas as suas ocorrências. Abaixo
está um exemplo do que você deve fazer.

Texto:
"Maria trabalha no Google em São Paulo desde 2022.\"

Resposta:
[
  {{ "entidade": "Maria", "tipo": "PESSOA" }},
  {{ "entidade": "Google", "tipo": "ORGANIZACAO" }},
  {{ "entidade": "São Paulo", "tipo": "LOCAL" }},
  {{ "entidade": "2022", "tipo": "DATA" }}
]

faça o mesmo com este texto no documento txt, e envie o json
resultante como ficheiro:
```

Figura A.2: Prompt utilizado com o Chat-GPT