UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ANDRIELLY DE LIMA LUCENA

UTILIZANDO EXTRAÇÃO DE RELAÇÃO ENTRE ENTIDADES PARA DETECÇÃO DE INFORMAÇÕES PESSOAIS SENSÍVEIS EM PORTUGUÊS

CAMPINA GRANDE - PB

2024

ANDRIELLY DE LIMA LUCENA

UTILIZANDO EXTRAÇÃO DE RELAÇÃO ENTRE ENTIDADES PARA DETECÇÃO DE INFORMAÇÕES PESSOAIS SENSÍVEIS EM PORTUGUÊS

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador : Fábio Jorge Almeida Morais

CAMPINA GRANDE - PB

2024

ANDRIELLY DE LIMA LUCENA

UTILIZANDO EXTRAÇÃO DE RELAÇÃO ENTRE ENTIDADES PARA DETECÇÃO DE INFORMAÇÕES PESSOAIS SENSÍVEIS EM PORTUGUÊS

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Fábio Jorge Almeida Morais

Orientador – UASC/CEEI/UFCG

Carlos Eduardo Santos Pires

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 15 de maio de 2024.

CAMPINA GRANDE - PB

RESUMO

Atualmente, a grande gama de plataformas, aplicativos e operações online disponíveis para a resolução de diferentes problemas resulta em um tráfego de grande volume de dados de usuários, inclusive dados sensíveis e de identificação. Para proteger a privacidade dos usuários, um direito assegurado por leis em todo o mundo (Leis de Proteção de Dados), é necessária uma atenção maior a esses dados para não serem publicados. No entanto, identificar as informações sensíveis entre tantos outros tipos de dados, pode não ser uma tarefa trivial. Estudos já existentes propõem a aplicação de técnicas de Processamento de Linguagem Natural (PLN) para identificação automática de Informações Pessoais Identificáveis (Personal Identifiable Information, PII) em documentos em português. O objetivo deste trabalho é propor, através de uma prova de conceito, uma abordagem complementar às utilizadas nos estudos relacionados, através da tarefa de Extração de Relação de PLN. Para tal, foi criado um componente que combina um modelo de linguagem especializado na língua portuguesa e camadas adicionais de extração de relação. Para o treinamento e avaliação do componente, foi gerada uma base de dados sensíveis sintéticos com o auxílio de um Large Language Model (LLM). Os resultados foram satisfatórios, com métricas de precisão, recall e f1-score acima de 95%, indicando que a abordagem pode ser uma boa proposta para detecção automática de informações sensíveis pessoais.

USING ENTITY RELATIONSHIP EXTRACTION FOR DETECTION OF SENSITIVE PERSONAL INFORMATION IN PORTUGUESE

ABSTRACT

Currently, the wide range of platforms, applications, and online operations available for solving different problems result in a high volume of user data traffic, including sensitive and identifying data. To protect users' privacy, a right guaranteed by laws worldwide (Data Protection Laws), greater attention to these data is necessary to prevent their disclosure. However, identifying sensitive information among many other types of data may not be a trivial task. Existing studies propose the application of Natural Language Processing (NLP) techniques for the automatic identification of Personal Identifiable Information (PII) in Portuguese documents. The aim of this work is to propose, through a proof of concept, a complementary approach to those used in related studies, through the task of NLP Relation Extraction. To do so, a component was created that combines a language model specialized in the Portuguese language and additional layers of relation extraction. For the training and evaluation of the component, a synthetic sensitive database was generated with the assistance of a Large Language Model (LLM). The results were satisfactory, with precision, recall, and f1-score metrics above 95%, indicating that the approach could be a good proposal for automatic detection of sensitive personal information.

Utilizando Extração de Relação entre entidades para detecção de informações pessoais sensíveis em português

Andrielly de Lima Lucena
Universidade Federal de Campina Grande
Campina Grande - PB
andrielly.lucena@ccc.ufcg.edu.br

Fábio Jorge Almeida Morais
Universidade Federal de Campina Grande
Campina Grande - PB
fabio@computacao.ufcg.edu.br

RESUMO

Atualmente, a grande gama de plataformas, aplicativos e operações online disponíveis para a resolução de diferentes problemas resulta em um tráfego de grande volume de dados de usuários, inclusive dados sensíveis e de identificação. Para proteger a privacidade dos usuários, um direito assegurado por leis em todo o mundo (Leis de Proteção de Dados), é necessária uma atenção maior a esses dados para não serem publicados. No entanto, identificar as informações sensíveis entre tantos outros tipos de dados, pode não ser uma tarefa trivial. Estudos já existentes propõem a aplicação de técnicas de Processamento de Linguagem Natural (PLN) para identificação automática de Informações Pessoais Identificáveis (*Personal Identifiable Information, PII*) em documentos em português. O objetivo deste trabalho é propor, através de uma prova de conceito, uma abordagem complementar às utilizadas nos estudos relacionados, através da tarefa de Extração de Relação de PLN. Para tal, foi criado um componente que combina um modelo de linguagem especializado na língua portuguesa e camadas adicionais de extração de relação. Para o treinamento e avaliação do componente, foi gerada uma base de dados sensíveis sintéticos com o auxílio de um *Large Language Model* (LLM). Os resultados foram satisfatórios, com métricas de precisão, *recall* e *f1-score* acima de 95%, indicando que a abordagem pode ser uma boa proposta para detecção automática de informações sensíveis pessoais.

Palavras-chave

PII, informação sensível, detecção, processamento de linguagem natural, extração de relação.

1. INTRODUÇÃO

Atualmente, existe um crescimento contínuo da quantidade de dados criados, capturados, copiados e consumidos digitalmente no mundo. Uma pesquisa realizada com informações de 2010 a 2020, mostra que esse número em 2020 foi de 64.2 Zettabytes, com uma projeção de aumento para todos os anos subsequentes até 2025 [1]. Parte desses dados são informações sensíveis e/ou pessoais que, baseando-se no direito à privacidade, devem ser manipulados com segurança para não se tornarem públicos.

Esse cenário trouxe a criação de legislações por todo o mundo que regulamentam o uso e armazenamento de informações sensíveis, como é o caso da Lei Geral de Proteção de Dados, n° 13.709/2018 [2]. Em 2023, houve o primeiro caso de aplicação de multa por descumprimento à LGPD no Brasil [3], e frequentemente, em todo o mundo, empresas sofrem indenizações por vazamento de dados [4].

Diante disto, é clara a importância da prudência nas operações que lidam com dados pessoais ou sensíveis. Para implementar medidas de segurança nessas operações, no entanto, é necessário detectar se as informações que estão sendo manipuladas são ou não de natureza sensível. Considerando a quantidade potencialmente grande de dados, é preciso ter técnicas computacionais automáticas para fazer essa detecção.

No entanto, essa não é uma tarefa trivial, já que nem sempre as informações se apresentarão de maneiras padronizadas e/ou estruturadas. É necessário um trabalho maior de Processamento de Linguagem Natural (PLN). Nesse caso, podem ser utilizadas técnicas como a de Reconhecimento de Entidade Nomeada (NER, do inglês Named Entity Recognition), que consiste em reconhecer em um texto palavras ou trechos que representam categorias predefinidas.

Estudos já foram realizados nessa área, como o de Dias et al. [5], que utiliza a tarefa de NER para detectar informações sensíveis, treinando modelos de PLN para reconhecer entidades que representam dados dessa natureza. No entanto, para alguns tipos de informações, apenas o reconhecimento de entidade pode não ser suficiente para determinar se ela é sensível ou não. Um endereço, por exemplo, pode ser reconhecido por ser uma entidade de local, mas o que o determina como um dado pessoal sensível é sua relação com alguma entidade de pessoa no texto.

Nessa problemática, o trabalho tem como objetivo, para a detecção de dados sensíveis em português, propor, através de uma prova de conceito, uma abordagem que aplica técnicas de PLN complementares ao NER, neste caso a Extração de Relação (Relation Extraction), que consiste em identificar relações entre as Entidades Nomeadas de um texto. Neste sentido, além de procurar por entidades de locais, dados médicos, informações de gênero, etc., também seria verificado se há relações dessas entidades com uma entidade de pessoa. Em caso positivo, trataria-se de um indicativo mais significativo da presença de um dado sensível pessoal.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Processamento de linguagem natural

O processamento de linguagem natural, PLN, é uma subárea da Inteligência Artificial que envolve processar e manipular dados não estruturados baseados em texto em linguagem humana. Esse

processamento é feito para extrair informações e características de certo texto, para analisar seus elementos ou gerar novos textos em linguagem natural a partir de outros. Atualmente a área engloba muitas tarefas computacionais que estão se desenvolvendo e crescendo cada vez mais.

2.2 Reconhecimento de Entidade Nomeada

Uma das tarefas do PLN, o Reconhecimento de Entidades Nomeadas, REN (ou NER, do inglês Named Entity Recognition), consiste na tarefa de identificar, dentro de um texto de um documento, as entidades presentes, categorizando em entidades predefinidas. Por exemplo, no texto "João Silva nasceu em 01/01/1980", sendo definidas previamente as entidades *Pessoa* e *Data*, um sistema de NER identificaria o trecho "João Silva" como *Pessoa* e "01/01/1980" como *Data*.

2.3 Extração de Relação

Também uma das tarefas de PLN, a extração de relação tem como objetivo identificar relações semânticas entre entidades de um texto. Para esta tarefa, também é necessário definir previamente as relações que serão extraídas. Por exemplo, em um texto: "João Silva, do Rio de Janeiro", "João Silva" seria categorizado como *Pessoa* e Rio de Janeiro como *Local*. Uma relação previamente definida de *nascer_em*, por exemplo, poderia ser atribuída a essas duas entidades para indicar a informação de que João Silva nasceu no Rio de Janeiro.

2.4 Arquitetura *Transformer*

A arquitetura *Transformer* [6] é uma arquitetura de redes neurais que se popularizou principalmente na área de PLN e se apresentou como uma alternativa às Redes Neurais Recorrentes, que são um tipo de arquitetura de rede neural projetada para lidar com dados sequenciais ou temporais.

Uma grande diferença entre as duas é que redes neurais com arquitetura *Transformer* têm uma grande capacidade de paralelização, devido ao fato de conseguir atribuir pesos diferentes a cada parte da entrada, sem necessariamente depender dos cálculos dos pesos das outras partes.

Essa atribuição de pesos é o chamado mecanismo de atenção. A ideia é associar pesos maiores a partes da entrada que são mais importantes para a tarefa desejada. Em um passo de uma tradução de um texto, por exemplo, será atribuído um peso maior para a palavra que está sendo traduzida no momento. Para isso, essa atribuição de pesos também leva em consideração o que já foi traduzido (o contexto da saída).

A arquitetura Transformer é composta por uma série de camadas, que empregam operações de atenção e transformações lineares. Existem as camadas de *encoders*, que são responsáveis por codificar a entrada, aplicando as operações matemáticas dos mecanismos de atenção e os *decoders*, que utilizam as representações intermediárias produzidas pelos *encoders* para gerar a saída final do modelo.

Essa arquitetura está representada de uma maneira simplificada na Figura 1. Como citado anteriormente, ela se baseia em camadas de *encoders* e *decoders*, em que ambas apresentam mecanismos de atenção para capturar as relações e importâncias nos textos de entrada e saída.

Essa estrutura de *encoders* e *decoders*, combinada com o mecanismo de atenção, permite que a arquitetura Transformer capture relações complexas em sequências de dados de maneira eficaz, tornando-a altamente versátil e eficiente em uma variedade de tarefas de NLP.

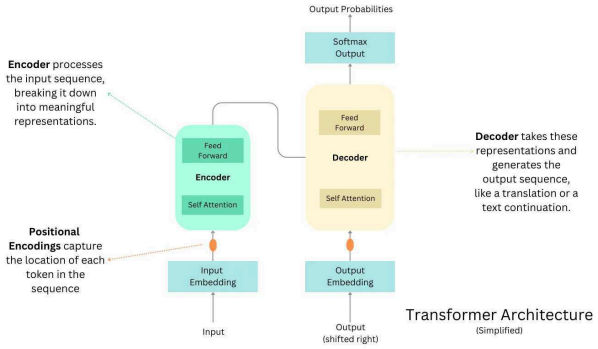

Figura 1: Arquitetura Transformer simplificada

Fonte: medium.com/@tech-gumptions

2.5 BERT

O modelo BERT, *Bidirectional Encoder Representation from Transformers* é um modelo de linguagem criado pela Google [7] e é baseado na arquitetura *Transformer*. Uma característica importante é o treinamento bidirecional, que, em vez de ser treinado para prever a próxima palavra de um texto, como era feito nos modelos anteriores a este, ele foi treinado para preencher lacunas dentro de um texto de acordo com o contexto da sentença, e para prever se uma sentença é subsequente a outra. Isso cria o aprendizado bidirecional, já que o contexto a ser levado em consideração não é só dos *tokens* (unidades básicas de texto) anteriores, mas do texto como um todo.

O modelo também é poderoso pelo fato de ter sido pré-treinado com uma quantidade muito grande de dados, beneficiando seu poder de generalização e seu desempenho em tarefas de processamento de linguagem natural. Isso favorece o ajuste fino (*fine tuning*) que pode ser feito para ajustar o modelo para tarefas específicas, através da adição de uma ou mais camadas que serão treinadas junto com o modelo para dados rotulados e mais específicos para determinada tarefa.

2.6 BERTimbau

BERTimbau [8] foi como foi chamada a versão BERT para o idioma português do Brasil. Essa versão foi treinada com uma grande quantidade de dados em português com o objetivo de compreender e processar o idioma em diversas nuances.

Assim como o BERT original, no BERTimbau também podem ser feitos ajustes para tarefas específicas, que é a proposta do presente trabalho, que utilizará essa versão do BERT para a tarefa de Extração de Relação.

3. TRABALHOS RELACIONADOS

Alguns trabalhos já foram realizados utilizando PLN para detecção de dados sensíveis em documentos em inglês. Yongyan Guo et al.[9] desenvolveram uma solução que utiliza tanto casamento de padrões com ReGex (para dados com formato padronizado, como e-mail), quanto um modelo treinado para a tarefa de NER (para dados que têm formatos menos definidos, como endereço). Os dados utilizados foram obtidos em

documentos do Pastebin[10], ferramenta que permite usuários postarem textos que ficam disponíveis publicamente. Para o modelo, foi utilizado uma combinação de técnicas de *Deep Learning* que chegaram aos resultados de 98.72% de precisão, 99.58% de *recall* e 99.15% de *f1-score*.

Já Fadi Hassan et al. [11] apresentou uma prova de conceito para anonimizar dados confidenciais utilizando a técnica de NER com *Conditional Random Fields* (CRF). O CRF é um tipo de modelo supervisionado probabilístico que extrai características dos *tokens* e associa essas características diretamente com as classes de saída através de pesos, sendo o peso da característica a importância dela em determinar cada classe de saída. Como prova de conceito, o trabalho focou em detectar nomes de doenças em textos de diagnósticos médicos. Os resultados foram 74.2% de precisão, 66% de *recall* e 69.8% de *f1-score*.

Mariana Dias et al. [5] também propuseram uma abordagem híbrida, utilizando casamento de padrões, algoritmos de *Machine Learning* e redes neurais para detectar diferentes tipos de dados sensíveis em documentos em português. Os dados utilizados foram de uma base anotada em português. A origem dos dados é diversa, por isso foi feita uma espécie de curadoria para determinar quais documentos continham informações consideradas sensíveis. A técnica de casamento de padrões foi utilizada apenas para uma parte das classes, enquanto para a outra foi comparada a utilização de modelos estatísticos e redes neurais. A abordagem de redes neurais obteve o melhor resultado, com *f1-score* de 83.01%, enquanto o melhor resultado dos modelos estatísticos obteve um *f1-score* de 65.50%.

Em abordagem complementar à dos trabalhos relacionados, o presente trabalho propõe a utilização de um modelo estado-da-arte em algumas tarefas de PLN em português, o BERTimbau, através da Extração de Relação. A diferença de abordagem em relação aos outros trabalhos citados é a adição de uma nova camada de análise do texto nos documentos para detectar relações entre os elementos, focado em textos da língua portuguesa.

4. METODOLOGIA

Neste trabalho foram desenvolvidas as etapas de um *pipeline* de desenvolvimento de solução utilizando Aprendizado de Máquina, considerando desde a obtenção e anotação de dados até treinamento e avaliação do modelo.

4.1 Geração da base de dados

Em se tratando do tópico de informações sensíveis, é previsível que não haja uma base de dados aberta e pública que possa ser utilizada para treinar e avaliar um modelo de *Machine Learning*. Por isso, para a etapa de obtenção de dados para o trabalho, foi feito o uso de um Large Language Model (LLM), modelo especialista em linguagem natural, para gerar textos com dados sensíveis e sintéticos.

Para a geração de tais textos, foi considerado o que a LGPD define como dado pessoal sensível no Art. 5° inciso II:

> *dado pessoal sensível: dado pessoal sobre origem racial ou étnica, convicção religiosa, opinião política, filiação a sindicato ou a organização de caráter religioso, filosófico ou político, dado referente à saúde ou à vida sexual, dado genético ou biométrico, quando vinculado a uma pessoa natural;*

Dada essa definição, o modelo de documento gerado foi o de relatório médico, que por si só pode conter alguns dos tipos de dado pessoal sensível citados na lei: origem étnica ou racial, dados relacionados à saúde e vida sexual e dados biométricos de uma pessoa.

Os textos foram gerados pelo modelo GPT-3-Turbo da OpenAI[12], através de sua API de comunicação. O prompt utilizado foi o descrito na Tabela 1.

Prompt utilizado para a geração de documentos
Dê exemplo de Registro Médico que contém informações de origem racial ou étnica, questões genéticas, biométricas e sobre a saúde ou a vida sexual de uma pessoa, utilizando dados sintéticos. o texto exemplo pode ser em formato de texto corrido ou de formulário. responda apenas com o exemplo pedido.

Tabela 1: Prompt utilizado para a geração de documentos

Um exemplo de saída da requisição está indicado na Tabela 2.

Resposta
Nome: João Silva *Data de nascimento: 01/01/1980* *Gênero: Masculino* *Origem racial ou étnica: Afrodescendente* *Histórico genético: Não apresenta predisposição genética para doenças hereditárias.* *Biométricas: Altura: 1,75m / Peso: 75kg / Grupo sanguíneo: O+* *Saúde: Diagnóstico médico de asma controlada com medicamentos. Nenhuma outra doença crônica.* *Vida sexual: Ativo sexualmente, sem histórico de DSTs.*

Tabela 2: Exemplo de resposta do GPT ao prompt

Foram obtidos 180 textos para serem utilizados no treinamento e avaliação do modelo.

4.2 Anotação da base de dados

Para que o modelo aprenda com os dados gerados, é necessário indicar nos textos quais são as entidades e quais as relações que existem entre elas, através de anotações. Assim, na fase de treinamento, o modelo detecta padrões para identificar tais entidades e relações. Já na fase de avaliação, o modelo compara sua resposta com a anotação, que seria o "gabarito".

Para a anotação dos textos, feita manualmente, foi utilizada a ferramenta *brat*[13], que disponibiliza uma interface gráfica que

permite selecionar o texto e indicar qual entidade aquele trecho representa, além de permitir indicar relações entre as entidades no texto. Um exemplo de uso da ferramenta está representado na Figura 2. Na imagem há um exemplo de texto com palavras destacadas, que representam as entidades. Na ligação entre as palavras relacionadas há a indicação de qual é a relação existente entre as entidades do texto.

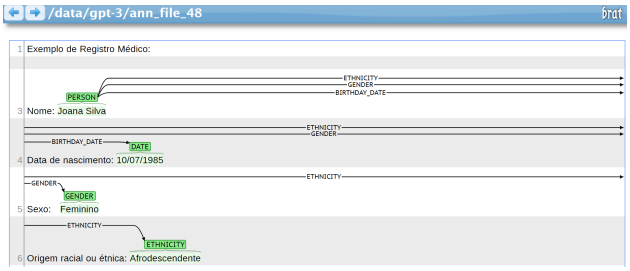

Figura 2: Ferramenta de Anotação

A Tabela 3 apresenta as entidades predefinidas para a anotação e experimentos do trabalho junto com uma descrição do que representa cada entidade.

Entidade	**Descrição**
PERSON	*Tokens* que correspondem a nomes próprios de pessoas
ADDRESS	*Tokens* que correspondem a endereços
GENDER	*Tokens* que correspondem a informações de gênero
ETHNICITY	*Tokens* que correspondem a informações de etnia ou raça
MEDICAL_INFO	*Tokens* que correspondem a informações médicas
BIOMETRIC	*Tokens* que correspondem a biométricas
SEXUAL_INFO	*Tokens* que correspondem a informações sexuais
DATE	*Tokens* que correspondem a datas

Tabela 3: Entidades predefinidas na anotação dos dados

A Tabela 4 apresenta as relações predefinidas para a anotação e experimentos do trabalho junto com uma descrição do que cada relação representa.

Relação	**Descrição**
PERSONAL_ADDRESS	Relação entre uma entidade de pessoa e uma entidade de endereço
BIRTHDAY_DATE	Relação entre uma entidade de pessoa e uma entidade de data, que represente a data de nascimento da pessoa
GENDER	Relação entre uma entidade de pessoa e uma entidade de gênero
ETHNICITY	Relação entre uma entidade de pessoa e uma entidade de etnia ou raça
MEDICAL_INFO	Relação entre uma entidade de pessoa e uma entidade de informação médica
BIOMETRIC	Relação entre uma entidade de pessoa e uma entidade de biométrica
SEXUAL_INFO	Relação entre uma entidade de pessoa e uma entidade de informação sexual

Tabela 4: Relações predefinidas

As anotações foram exportadas da ferramenta em um arquivo de anotação do tipo *ann*. Foram necessárias conversões de formatos para adaptar os dados anotados a um formato que o componente de extração de Relação reconhecesse. Como foi utilizado a biblioteca *Spacy*[14] para o treinamento e avaliação do modelo, os dados precisaram ser passados para o tipo de arquivo *SPACY*. O arquivo de anotações foi tratado em um *script Python*[15] para ser transformado em um formato *JSON*, e depois em outro *script*[16], para ser convertido em um formato *SPACY* que a biblioteca reconhece.

4.3 Treinamento

As execuções dos passos de treinamento e de avaliação foram feitas utilizando o *Google Colab*[17], com a configuração T4: 12.7GB de RAM, 15 GB de GPU RAM e 78.2 GB de armazenamento.

O treinamento do modelo foi baseado no código disponibilizado pelo *Spacy*[18], que tem suporte para treinar qualquer modelo com arquitetura *Transformers* disponível no *HuggingFace*[19]. O módulo faz uso de um arquivo de configurações que define alguns parâmetros de execução e os hiperparâmetros dos componentes a serem treinados. Foi modificada a configuração de linguagem, para utilizar o português, e a configuração do modelo pré-treinado a ser utilizado (neste caso, o BERTimbau). Para esta etapa, foram utilizados 70% dos documentos gerados como dados para o componente.

Para a análise do modelo durante a fase de treinamento foram considerados 15% dos documentos. A cada intervalo de passos do processo (o intervalo é definido no arquivo de configuração) as métricas do modelo eram calculadas de acordo com esse conjunto de dados.

A arquitetura utilizada para o componente de extração de relação com o BERTimbau está representada na Figura 3. O BERTimbau foi utilizado para gerar os vetores de representação das palavras do documento dado como entrada. Esses vetores que são as saídas do modelo passarão para uma camada

adicional de *reduce_mean*, que calcula um vetor que será a média de todos os vetores da saída do BERT. O vetor-média é concatenado com os vetores de representação dos *tokens* das entidades entre as quais se quer prever uma relação, produzindo um vetor único que contém informações do contexto geral do documento e especificamente das entidades presentes. O resultado da concatenação passa para uma camada de classificação, que retorna um valor entre 0 e 1 para cada possível relação presente no texto, e esse valor representa a probabilidade de a relação estar presente entre as duas entidades em questão.

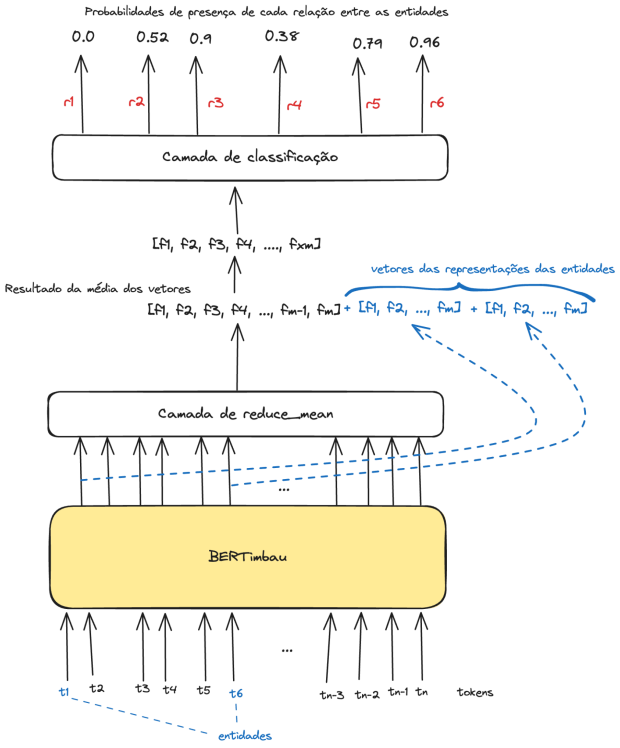

Figura 3: Representação da arquitetura utilizada

Durante o treinamento, são calculadas 5 métricas usadas para avaliar o modelo: perda (ou *loss*) do componente *Transformer*, nesse caso o BERT; perda do componente de Extração de Relação, que corresponde ao conjunto de camadas adicionais descritas na seção anterior; a precisão micro; *recall* micro e *f1-score* micro. As perdas correspondem às métricas de erro, remetem ao quão distantes estão as previsões do modelo dos rótulos verdadeiros. O objetivo do treinamento é minimizar essas métricas.

Por outro lado, as métricas de precisão, *recall* e *f1-score* correspondem às proporções de acerto do modelo. Entendendo-se como exemplo positivo a presença de uma relação X entre duas entidades, a precisão é a proporção de exemplos positivos previstos corretamente em relação ao total de exemplos previstos como positivos. A *recall* é a proporção de exemplos positivos previstos corretamente em relação ao total de exemplos positivos no conjunto de dados. O *f1-score* é uma métrica que combina precisão e *recall* em uma única medida, calculada como a média harmônica das duas.

A versão micro de cada uma dessas métricas é a média ponderada dos seus valores para cada classe (nesse caso, para cada relação), pesada pelo número de exemplos em cada classe. Métricas micro são especialmente úteis quando há um desbalanceamento entre as classes nos dados, pois dão mais peso às classes com mais exemplos.

4.4 Avaliação

Para a etapa de avaliação, foram utilizados os 15% restantes dos documentos. Neste passo, é dado ao modelo treinado um texto com indicações de quais são as entidades presentes nele e o modelo deve inferir se existem e quais são as relações entre as entidades.

Nesta etapa, as métricas são calculadas em função de um limite numérico (*threshold*) que é definido para indicar se a relação está presente ou não. Por exemplo: se o limite, ou *threshold,* da presença de uma relação é de 0.5, apenas relações que tiveram probabilidade maior que 0.5 na camada de classificação serão consideradas como presentes entre as entidades. A fins de comparação, o modelo treinado é avaliado com diferentes *thresholds*, variando de 0.0 a 1.0, como mostra a Figura 4.

```
Results of the trained model:
threshold 0.00   {'rel_micro_p': '1.83', 'rel_micro_r': '100.00', 'rel_micro_f': '3.59'}
threshold 0.05   {'rel_micro_p': '17.35', 'rel_micro_r': '100.00', 'rel_micro_f': '29.57'}
threshold 0.10   {'rel_micro_p': '36.88', 'rel_micro_r': '100.00', 'rel_micro_f': '53.89'}
threshold 0.20   {'rel_micro_p': '74.12', 'rel_micro_r': '100.00', 'rel_micro_f': '85.14'}
threshold 0.30   {'rel_micro_p': '96.67', 'rel_micro_r': '100.00', 'rel_micro_f': '98.31'}
threshold 0.40   {'rel_micro_p': '98.28', 'rel_micro_r': '98.28', 'rel_micro_f': '98.28'}
threshold 0.50   {'rel_micro_p': '99.56', 'rel_micro_r': '97.41', 'rel_micro_f': '98.47'}
threshold 0.60   {'rel_micro_p': '100.00', 'rel_micro_r': '95.26', 'rel_micro_f': '97.57'}
threshold 0.70   {'rel_micro_p': '100.00', 'rel_micro_r': '91.38', 'rel_micro_f': '95.50'}
threshold 0.80   {'rel_micro_p': '100.00', 'rel_micro_r': '80.60', 'rel_micro_f': '89.26'}
threshold 0.90   {'rel_micro_p': '100.00', 'rel_micro_r': '59.05', 'rel_micro_f': '74.25'}
threshold 0.99   {'rel_micro_p': '0.00', 'rel_micro_r': '0.00', 'rel_micro_f': '0.00'}
threshold 1.00   {'rel_micro_p': '0.00', 'rel_micro_r': '0.00', 'rel_micro_f': '0.00'}
```

Figura 4: Logs de avaliação do modelo treinado

5. RESULTADOS

Inicialmente, foi executado o treinamento com os hiperparâmetros padrões do arquivo de configuração. No entanto, foi notado um indício de *overfitting* do modelo, dado que o número de passos sendo executados era muito grande para a quantidade de dados disponíveis. Então, o modelo estava se especializando muito apenas nos dados fornecidos, o que compromete o seu poder de generalização. A curva de aprendizado dessa execução pode ser vista na Figura 5. O valor das perdas no treinamento tem uma queda abrupta entre o 100° e o 200° passo, o que pode indicar uma especialização muito alta nos dados de treino a partir deste passo.

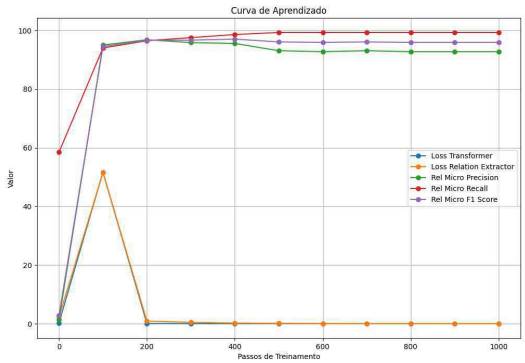

Figura 5: Métricas de treinamento da primeira execução

Com base nisso, foi feito um conjunto de experimentos para análise da variação de dois hiperparâmetros importantes no processo de treinamento de um modelo: o *batch size* e o número

máximo de passos. O *batch size,* ou tamanho do lote, refere-se à quantidade de exemplos que são processados em paralelo durante o treinamento. Um *batch size* maior implica em um treinamento mais rápido, já que mais dados são processados paralelamente, mas também requer uma quantidade maior de memória disponível para esses processamentos.

Por outro lado, o número de passos se refere à quantidade de alterações nos pesos da rede para minimizar o erro durante o treinamento, que geralmente ocorre após o processamento de um lote de dados. Um número pequeno de passos pode resultar em uma convergência precoce e impedir a rede de aprender com todo o conjunto de dados disponível. Já um número de passos muito grande pode levar a excessivas iterações sobre os exemplos, fazendo o modelo memorizar os dados em vez de aprender padrões gerais.

Desta forma, foram definidos 9 cenários para o treinamento do componente de Extração de Relação, cada um com uma combinação diferente de valores para o número de passos e *batch size*. A descrição dos cenários em relação aos valores dos hiperparâmetros está representada na Tabela 5.

	batch_size	**número de passos**
Cenário 1	500	250
Cenário 2	500	500
Cenário 3	500	1000
Cenário 4	1000	250
Cenário 5	1000	500
Cenário 6	1000	1000
Cenário 7	2000	250
Cenário 8	2000	500
Cenário 9	2000	1000

Tabela 5: Hiperparâmetros dos cenários de experimentos

Para cada cenário, foram calculadas as métricas micro de precisão, *recall* e *f1-score*, variando o *threshold* de 0.00 a 1.00. Os valores das métricas para cada *threshold* estão representados respectivamente nos gráficos das Figuras 6, 7, e 8.

Para a métrica de precisão, é possível notar na Figura 6 que, em todos os cenários, valores de *threshold* a partir de 0.4 e 0.6 são os que apresentam melhores resultados. Isso se deve ao fato de que um *threshold* baixo significa ser mais brando em relação a presença de uma relação entre entidades, ou seja, mesmo com uma baixa probabilidade de certa relação existir, o modelo considera a presença dela. Tal comportamento aumenta o número de falsos positivos, que influencia negativamente na métrica de precisão.

Já para a métrica de micro *recall*, como mostra a Figura 7, os valores são altos desde o primeiro valor de *threshold* e só caem a partir de 0.9. Isso acontece pois valores menores para o *threshold* implicam em considerar mais exemplos como positivos em relação à presença de uma relação, e isso aumenta o valor do *recall*, já que a chance de se considerar corretamente um exemplo como positivo também aumenta. Vale ressaltar que, neste contexto de dados sensíveis pessoais, é prudente priorizar a métrica de *recall*, já que ter mais falsos positivos apresenta uma gravidade menor do que ter mais falsos negativos.

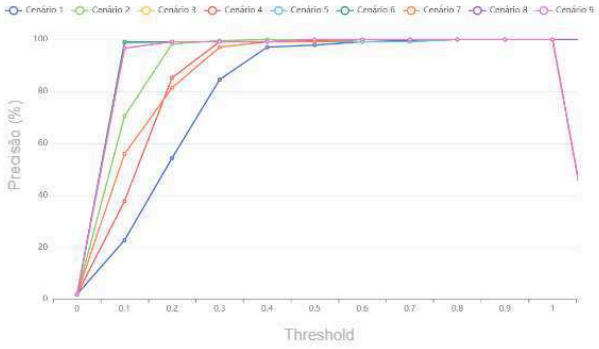

Figura 6: Micro precisão de todos os cenários em função do *threshold*

Para a métrica de *f1-score*, como mostrado na Figura 8, valores de *threshold* mais centrais são os que maximizam a métrica. Como o *f1-score* é uma média harmônica entre as duas métricas citadas acima, faz sentido analisá-la na tomada de decisão para definir valores que maximizem o conjunto das duas métricas.

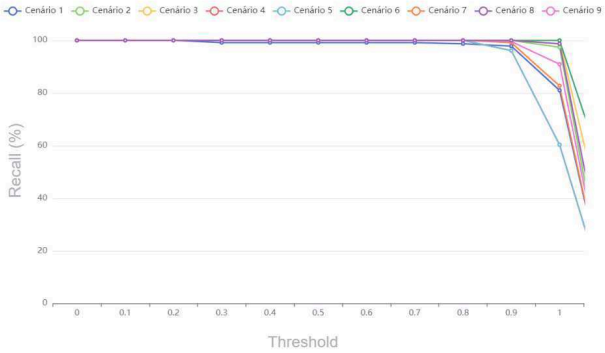

Figura 7: Micro *recall* de todos os cenários em função do threshold

Assim, é possível perceber que os cenários mostraram resultados parecidos em um intervalo central de valores de *threshold*. Além disso, diante do que foi citado, são preferíveis os menores valores que maximizam as métricas, já que estes têm uma influência positiva na métrica de *recall*, que deve ser priorizada neste contexto.

6. CONCLUSÃO

Neste trabalho foi proposto uma prova de conceito utilizando a tarefa de Extração de Relação em PLN para detectar informações pessoais sensíveis em documentos. Com os dados utilizados, gerados com o auxílio de um LLM, os resultados foram satisfatórios para iniciar a discussão e análise da utilização dessa abordagem em contextos de cibersegurança.

Os resultados também indicam que a utilização de um modelo já ajustado para a língua portuguesa facilita na interpretação e na

captura acurada do contexto e de nuances da língua, que fazem diferença para a classificação de relação entre palavras.

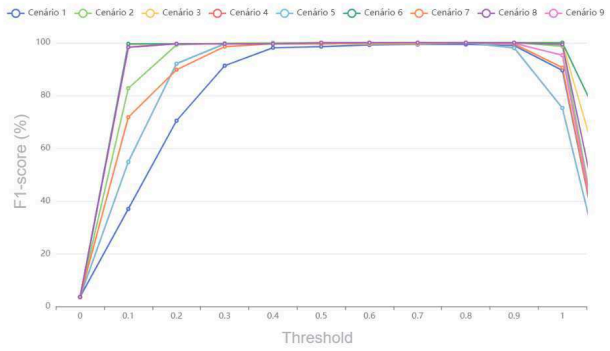

Figura 8: Micro *f1-score* de todos os cenários em função do *threshold*

Para trabalhos futuros, propomos o aumento da base de dados, para incluir diferentes modelos de texto, além de diferentes tipos de entidades e relações, para que o componente de detecção consiga entender diferentes contextos e aumente seu poder de generalização. Além disso, também é válida a análise dos resultados treinando o modelo com outros valores para os hiperparâmetros citados, neste caso utilizando valores menores, considerando a base de dados pequena. A variação de outros aspectos, como a própria arquitetura da rede ou o hiperparâmetro de janela de contexto também podem influenciar no seu desempenho e acurácia e podem ser considerados para trabalhos futuros.

Como limitações do trabalho, pode ser citada a base de dados utilizada. Por não ter sido possível utilizar dados reais para o treinamento do modelo, os dados precisaram ser gerados. Essa base de dados gerada tem limitações no sentido de ser restrita a um certo contexto e possivelmente não ser representativa o suficiente para obter um componente de detecção com uma característica de generalização satisfatória. O uso de bases de dados mais abrangentes podem ajudar a mitigar essa limitação.

7. AGRADECIMENTOS

Eu não poderia ter chegado até aqui sozinha, por isso não posso deixar de agradecer às muitas pessoas que fizeram parte dessa minha caminhada na graduação.

A toda a comunidade de Ciência da Computação na UFCG: discentes, docentes, servidores e técnico-administrativos, por manterem a excelência do curso, em especial a Fábio Morais pelas orientações em TCC, disciplinas e projetos.

A minha família: meus pais Antonio Lucena e Adalvaneide de Lima, por sempre acreditarem na minha educação e confiarem nas minhas escolhas. Ao meu irmão Anthonny por sempre celebrar minhas conquistas e me fazer querer dar o meu melhor.

A Andressa Lucena e Helen Cavalcanti, com quem dividi apartamento e importantes momentos durante esses anos, por sempre estarem física e emocionalmente ao meu lado, celebrando, estudando, conversando e convivendo. Não consigo mensurar a importância de vocês para mim.

Ao meu noivo Nícolas Moreira, com quem compartilho sentimentos e momentos de todos os tipos, por todo o tempo de qualidade, por sempre se preocupar em me confortar, mesmo passando pelas mesmas frustrações e por sempre saber o melhor para falar e fazer.

Ao maravilhoso acaso da união em um grupo de amigas com Sheila Paiva, Anna Beatriz e Helen Cavalcanti. Vocês foram minha constante rede de confidentes. Junto a elas, agradeço a Arthur Macena, Davi Sousa, Kennedy Dantas e Mariane Zeitouni por todos os momentos. A vida é muito mais legal com vocês.

A Arthur Alves, Emilly Albuquerque, Igor Farias, Mateus Ribeiro, Natalia Salvino, Antonio Bertino, Ricardo Adley e Henrique Lemos, pelas incontáveis horas de estudos, conversas, e diversão. Tenho muita sorte de ter podido contar com vocês em uma época de tantas incertezas para todos.

8. REFERÊNCIAS

[1] Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025 https://www.statista.com/statistics/871513/worldwide-data-created/

[2] LEI Nº 13.709, DE 14 DE AGOSTO DE 2018 https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709compilado.htm

[3] ANPD aplica a primeira multa por descumprimento à LGPD https://www.gov.br/anpd/pt-br/assuntos/noticias/anpd-aplica-a-primeira-multa-por-descumprimento-a-lgpd

[4] 8 casos de vazamentos de dados tratados com a LGPD https://www.softwall.com.br/blog/vazamentos-de-dados-tratados-com-a-lgpd/

[5] Dias, M.; Boné, J.; Ferreira, J.C.; Ribeiro, R.; Maia, R. Named Entity Recognition for Sensitive Data Discovery in Portuguese. Appl. Sci. 2020, 10, 2303. https://doi.org/10.3390/app10072303

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser e Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, e Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[8] Fábio Souza, Rodrigo Nogueira e Roberto Lotufo. 2020. BERTimbau: Pretrained BERT Models for Brazilian Portuguese. In Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part I. Springer-Verlag, Berlin, Heidelberg, 403–417. https://doi.org/10.1007/978-3-030-61377-8_28

[9] Yongyan Guo, Jiayong Liu, Wenwu Tang, Cheng Huang, Exsense: Extract sensitive information from unstructured data, Computers & Security, Volume 102, 2021, 102156, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2020.102156

[10] PASTEBIN https://pastebin.com/

[11] Fadi Hassan, Josep Domingo-Ferrer, and Jordi Soria-Comas. 2018. Anonymization of Unstructured Data via Named-Entity Recognition. In Modeling Decisions for Artificial Intelligence: 15th International Conference, MDAI 2018, Mallorca, Spain, October 15–18, 2018, Proceedings. Springer-Verlag, Berlin, Heidelberg, 296–305. https://doi.org/10.1007/978-3-030-00202-2_24
[12] OpenAI https://openai.com/
[13] brat rapid annotation tool https://brat.nlplab.org/
[14] spaCy · Industrial-strength Natural Language Processing in Python https://spacy.io/
[15] Script formatação *ann* para *JSON* https://github.com/andriellyll/portuguese-relation-extraction/blob/main/ann_to_json.py
[16] Script formatação *JSON* para *SPACY* https://github.com/andriellyll/portuguese-relation-extraction/blob/main/parse_data.py
[17] Google Colab https://colab.google/
[18] Weasel Project: Example project of creating a novel nlp component to do relation extraction from scratch. https://github.com/explosion/projects/tree/v3/tutorials/rel_component
[19] Hugging Face - The AI community building the future. https://huggingface.co/