UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO CIÊNCIAS DA EDUCAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA INFORMAÇÃO

Rodrigo Alves da Fonseca

Uma proposta de pipeline semiautomático baseado em verbalizações para extrair relacionamentos adequados entre entidades no contexto da atividade de inteligência e investigação policial de Pessoas Expostas Politicamente

Florianópolis
2024

Rodrigo Alves da Fonseca

Uma proposta de pipeline semiautomático baseado em verbalizações para extrair relacionamentos adequados entre entidades no contexto da atividade de inteligência e investigação policial de Pessoas Expostas Politicamente

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Informação do Centro de Ciências da Educação da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Ciência da Informação.

Orientador: Prof. Dr. Moisés Lima Dutra

Florianópolis
2024

Ficha catalográfica elaborada pela Bibliotecária Maria Angela Grechaki Dominhaki no site: http://portalbu.ufsc.br/ficha

Fonseca, Rodrigo Alves da
Uma proposta de pipeline semiautomático baseado em verbalizações para extrair relacionamentos adequados entre entidades no contexto da atividade de inteligência e investigação policial de Pessoas Expostas Politicamente / Rodrigo Alves da Fonseca ; orientador, Moisés Lima Dutra, 2024.
121 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro de Ciências da Educação, Programa de Pós-Graduação em Ciência da Informação, Florianópolis, 2024.

Inclui referências.

1. Ciência da Informação. 2. Processamento de Linguagem Natural. I. Dutra, Moisés Lima. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Informação. III. Título.

Rodrigo Alves da Fonseca

Uma proposta de pipeline semiautomático baseado em verbalizações para extrair relacionamentos adequados entre entidades no contexto da atividade de inteligência e investigação policial de Pessoas Expostas Politicamente

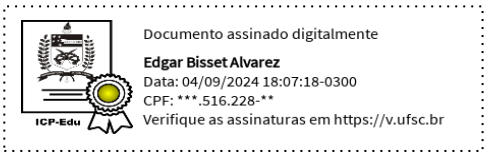
O presente trabalho em nível de Mestrado foi avaliado e aprovado , em 07 de agosto de 2024, pela banca examinadora composta pelos seguintes membros:

Prof.(a) Gustavo Medeiros de Araújo, Dr(a).
Universidade Federal de Santa Catarina

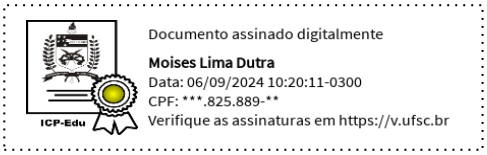
Prof.(a) Thiago Magela Rodrigues Dias, Dr(a).
Universidade Federal de Santa Catarina

Prof.(a) Sandro Rautenberg, Dr(a).
Universidade Estadual do Centro-Oeste

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Mestre em Ciência da Informação.



Coordenador(a) do Programa



Prof. Dr. Moisés Lima Dutra

Orientador(a)

Florianópolis, 07 de agosto de 2024.

Este trabalho é dedicado aos meus pais, Nilza e Zulmiro.

AGRADECIMENTOS

Devo meus agradecimentos a vários colaboradores, sem os quais este trabalho não poderia ter sido realizado. Primeiramente, agradeço ao meu amigo e chefe, Fabrício Bispo, pela colaboração, parceria e incentivo, proporcionando todas as condições necessárias para a realização deste trabalho.

Agradeço também ao colega Fillipe Rodrigues, que pacientemente me transmitiu conhecimentos técnicos indispensáveis, ao meu amigo José Herlânio pelo apoio e orientação, e ao amigo e incentivador Alcides Macêdo, por onde tudo isso começou.

Um agradecimento especial ao meu orientador, Prof. Moisés, pelas muitas horas dedicadas tanto à orientação quanto à revisão dos textos produzidos. Seu alto nível de exigência foi essencial para a realização deste trabalho.

À minha família, expresso um agradecimento muito especial. Durante o curso do mestrado, tivemos a honra de receber nosso filho caçula, Thiago. Devido ao esforço necessário para a produção desta dissertação, muitas vezes precisei privá-lo de tempo precioso quando ele mais precisava. Aos meus filhos, Milena e Arthur, agradeço a paciência e compreensão, pois muitas vezes deixei de atendê-los para trabalhar na produção textual. Um agradecimento mais que especial à minha esposa e companheira, Valéria. Sua dedicação integral aos filhos e à casa durante o período em que estive ausente foi fundamental para o sucesso deste trabalho.

Agradeço, por fim, a todos os demais professores e avaliadores que contribuíram para o aperfeiçoamento deste trabalho.

“A melhor parte de uma viagem é o caminho, não o destino.” (autor desconhecido)

RESUMO

Este trabalho investiga a extração de relacionamentos relevantes entre entidades envolvendo Pessoas Expostas Politicamente (PEPs) a partir de fontes abertas, com o objetivo de apoiar atividades de inteligência e investigação policial. Para isso, foi proposto um *pipeline* semiautomático de mineração textual baseado em verbalizações destinado a extrair relacionamentos da *Web* no contexto das atividades mencionadas. O primeiro passo envolveu a definição de um cenário de trabalho e a investigação de tarefas de Processamento de Linguagem Natural (PLN) necessárias para o desenvolver o *pipeline*. A extração de relacionamentos por meio de verbalizações opera testando hipóteses por meio da utilização de um modelo pré-treinado de inferência de PLN. Nesse processo, o texto a ser analisado é tratado como uma premissa, enquanto a verbalização é considerada a hipótese a ser testada. O pipeline utiliza um *corpus* textual anotado manualmente, extraído da *Web*, por meio de pesquisas de palavras-chave predefinidas, composto por documentos específicos relacionados às PEPs. Este *corpus* serve de base para o desenvolvimento das verbalizações. Os resultados indicam que o método é eficaz na identificação de relacionamentos entre entidades. O uso do modelo DeBERTa da *Microsoft*, com um limiar de negativação estabelecido em 0,8, apresentou melhores resultados na identificação de relacionamentos em *corpora* textuais relacionados ao domínio proposto. Em um cenário de classificação binária, foi possível obter um *F1-score* de 0,865 e acurácia de 0,967. A metodologia proposta destaca a importância das verbalizações no processo de extração de relacionamentos. Verbalizações inadequadas podem comprometer a qualidade da extração, tornando essa etapa crítica. Embora a técnica não necessite de treinamento de modelo, a escolha das verbalizações é crucial: aquelas muito específicas podem causar *overfitting*, enquanto as excessivamente gerais podem resultar em *underfitting*. Portanto, é essencial criar verbalizações baseadas em documentos do domínio específico de interesse, ajustando-as por meio de testes e refinamentos sucessivos.

Palavras-chave: Investigação Policial. Pessoas Expostas Politicamente. Processamento de Linguagem Natural. Extração de Relacionamentos.

ABSTRACT

This work investigates the extraction of relevant relationships between entities involving Politically Exposed Persons (PEPs) from open sources, with the aim of supporting intelligence and police investigation activities. To this end, a semi-automatic textual mining pipeline based on verbalizations was proposed to extract relationships from the Web in the context of the mentioned activities. The first step involved defining a work scenario and investigating Natural Language Processing (NLP) tasks necessary to develop the pipeline. Extracting relationships through verbalizations operates by testing hypotheses through the use of a pre-trained NLP inference model. In this process, the text to be analyzed is treated as a premise, while the verbalization is considered the hypothesis to be tested. The pipeline uses a manually annotated textual corpus, extracted from the Web, through predefined keyword searches, composed of specific documents related to PEPs. This corpus serves as the basis for the development of verbalizations. The results indicate that the method is effective in identifying relationships between entities. The use of Microsoft's DeBERTa model, with a negative threshold set at 0.8, showed better results in identifying relationships in textual corpora related to the proposed domain. In a binary classification scenario, it was possible to obtain an F1-score of 0.865 and an accuracy of 0.967. The proposed methodology highlights the importance of verbalizations in the relationship extraction process. Inadequate verbalization can compromise the quality of the extraction, making this step critical. Although the technique does not require model training, the choice of verbalizations is crucial: those that are too specific can lead to overfitting, while those that are too general can result in underfitting. Therefore, it is essential to create verbalizations based on documents from the relevant domain, adjusting them through successive tests and refinements.

Keywords: *Police investigation. Politically Exposed Person. Natural Language Processing. Relationship Extraction.*

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
BACEN	Banco Central do Brasil
BERT	Bidirectional Encoder Representations from Transformers
CAC	Campo Aleatório Condicional
CGU	Controladoria-Geral da União
CI	Ciência da Informação
CNJ	Conselho Nacional de Justiça
CNMP	Conselho Nacional do Ministério Público
COAF	Conselho de Controle de Atividades Financeiras
CVM	Comissão de Valores Mobiliários
DEN	Desambiguação das Entidades Nomeadas
EI	Extração de Informações
ENCCLA	Estratégia Nacional de Combate à Corrupção e à Lavagem de Dinheiro
ER	Extração de Relacionamentos
FEBRABAN	Federação Brasileira de Bancos
FPCC/RS	Fórum de Prevenção e Enfrentamento à Corrupção e à Improbidade Administrativa do Rio Grande do Sul
GPU	Graphic Processing Unit
HMM	Hidden Markov Model
KYC	Know Your Customer
LOC	Entidade Local
LSTM	Long Short-Term Memory
MEMM	Maximum-Entropy Markov Model
MISC	Entidades Diversas
MLM	Masked Language Model
MPF	Ministério Público Federal
MPOG	Ministério do Planejamento
MT	Mineração de Textos
NLI	Natural Language Inference
NLG	Natural Language Generation

NLU	Natural Language Understanding
NSP	Next Sentence Prediction
OIE	Open Information Extraction
ORG	Entidade Organização
PEP	Pessoas Expostas Politicamente
PES	Entidade Pessoa
PF	Polícia Federal
PLN	Processamento de Linguagem Natural
PREVIC	Superintendência Nacional de Previdência Complementar
Q&A	Questions and Answers
REN	Reconhecimento de Entidades Nomeadas
RFB	Receita Federal do Brasil
RSL	Revisão Sistemática da Literatura
SEGEP	Secretaria de Gestão de Pessoas do Ministério da Economia
SGBD	Sistema Gerenciador de Banco de Dados
STF	Supremo Tribunal Federal
SUSEP	Superintendência de Seguros Privados
TCU	Tribunal de Contas da União
VE	Vinculação de Entidade

SUMÁRIO

1 INTRODUÇÃO

No âmbito das atividades de Polícia Judiciária e nas ações de inteligência, identificar entidades e relacionamentos tem se tornado crucial para investigações e produção de conhecimento. Atualmente, muitas dessas informações valiosas estão dispersas em fontes abertas, não necessariamente registradas em bancos de dados governamentais, mas em conteúdos como mídias sociais e outras plataformas online. Com a popularização das mídias sociais, observa-se um aumento exponencial na geração de dados textuais diários, provenientes de plataformas como X (antigo *Twitter*), *Facebook*, *YouTube*, *Instagram*, além de fontes mais tradicionais como *Wikipedia* e portais de notícias. Esse cenário ressalta a relevância crescente do Processamento de Linguagem Natural (PLN), uma área de pesquisa que busca facilitar a interação entre a linguagem natural dos humanos e as máquinas. O PLN é essencial para análise e compreensão desses dados, sendo utilizado em diversas áreas, desde o mercado financeiro até as atividades de inteligência, onde profissionais precisam extrair informações relevantes de grandes volumes de texto. A produção de conhecimento como atividade de inteligência é uma atividade antiga e corriqueira em diversas esferas da sociedade, voltada para a obtenção de informações que possam ser aplicadas a propósitos específicos ou embasar processos decisórios.

Uma das tarefas fundamentais do PLN é o Reconhecimento de Entidades Nomeadas (REN), que visa identificar entidades como pessoas, organizações e locais em textos de formatos livres (não estruturados e semiestruturados). Além disso, não se trata apenas de reconhecer a presença das entidades, mas também de identificar as suas localizações exatas no texto. Essa identificação precisa é necessária para outras tarefas, como por exemplo para a Extração de Relacionamentos (ER). A ER busca identificar as relações entre duas ou mais entidades dentro de uma sentença, permitindo uma compreensão mais profunda das interações presentes nos dados textuais. Os relacionamentos entre entidades investigadas muitas vezes não são de amplo conhecimento da sociedade, e um determinado órgão de persecução criminal pode não ter tido acesso a documentos específicos sobre um caso em outro órgão. Ademais, também podem existir associações criminosas ou conexões informais que se mostrem fundamentais para desvendar crimes, especialmente no âmbito da corrupção e lavagem

de dinheiro, onde há uma estreita relação entre indivíduos e organizações expostas politicamente.

Avanços recentes em técnicas de aprendizado de máquina, como os modelos baseados em Transformadores, têm impulsionado significativamente o campo do PLN. O surgimento do BERT (*Bidirectional Encoder Representations from Transformers*), por exemplo, revolucionou a maneira como os modelos de linguagem são treinados e aplicados. Além disso, o conceito de aprendizado por transferência tem sido amplamente adotado, permitindo reutilizar o conhecimento adquirido em uma tarefa para outras, reduzindo o tempo e os recursos necessários para o treinamento de novos modelos.

Nesse contexto, a oferta de dados sobre Pessoas Expostas Politicamente (PEP), disponibilizados pela Controladoria-Geral da União (CGU), e a exigência de autodeclarações sobre status de PEP em formulários públicos e privados destacam a importância dessas informações para direcionar investigações e atividades de inteligência. Ademais, nos últimos anos, surgiram várias estratégias e abordagens que têm contribuído para melhorar o desempenho do PLN, incluindo o *Zero-Shot*, uma técnica de classificação de relacionamentos que se baseia na capacidade de transferência de aprendizado sem a necessidade de ajuste fino, utilizando uma tarefa de vinculação ou implicação textual. Essas novas técnicas permitem uma análise mais aprofundada e eficaz dos dados disponíveis, gerando conhecimentos valiosos para diversas aplicações.

No entanto, o PLN enfrenta desafios significativos, como a contextualização de palavras e a variedade de significados que podem assumir em diferentes contextos. Podemos dizer que essa dificuldade foi reduzida utilizando-se técnicas mais sofisticadas que surgiram ao longo do tempo, incluindo a combinação de grandes recursos de corpora textuais com técnicas estatísticas e linguísticas. Entretanto, a variedade e quantidade de textos em fontes abertas é ainda uma dificuldade que sempre deverá ser enfrentada com o desenvolvimento e aprimoramento de novas técnicas.

Diante desse contexto de grande disponibilidade de material em fontes abertas, esse trabalho procura responder: É possível extrair relacionamentos entre entidades num contexto de crimes de Pessoas Expostas Politicamente através de PLN para apoio da atividade de inteligência e investigação policial?

2 OBJETIVOS

Neste tópico serão delineados os objetivos: geral e os específicos desta pesquisa, considerando o contexto exposto anteriormente.

2.1 OBJETIVO GERAL

Propor um processo semiautomático de mineração textual baseado em verbalizações para, a partir de dados coletados de fontes abertas na *Web*, identificar relações significativas e relevantes entre entidades nomeadas reconhecidas no âmbito da atividade de inteligência e investigação policial envolvendo Pessoas Expostas Politicamente.

2.2 OBJETIVOS ESPECÍFICOS

a) Mapear elementos do Reconhecimento de Entidades Nomeadas (REN) e da Extração de Relacionamentos (ER) que sejam úteis para a pesquisa;
b) Estruturar os elementos de suporte da proposta;
c) Desenvolver o pipeline semiautomático para ER baseado em verbalizações;
d) Aplicar a proposta num cenário de teste de investigação policial.

2.3 JUSTIFICATIVA

A segurança pública e a persecução criminal são muito dependentes das medidas preventivas construídas em um país. Nesse contexto, o Brasil, acompanhando outros países, iniciou medidas de combate à corrupção e à lavagem de dinheiro, abrangendo um amplo espectro de instituições dos poderes Executivo, Legislativo e Judiciário das esferas federal, estadual e municipal, bem como dos Ministérios Públicos dos estados e da União. Em 2003, foi criada a Estratégia Nacional de Combate à Corrupção e à Lavagem de Dinheiro (ENCCLA[1]), que reúne mais de 70 dessas instituições para discutir, formular e

[1] https://enccla.camara.leg.br/

concretizar políticas públicas e soluções para enfrentar a corrupção e a lavagem de dinheiro (BRASIL, 2021a).

Durante esses 21 anos de existência, a ENCCLA se esforçou para que seus entes, em um ambiente de discussões voltadas ao aperfeiçoamento do sistema de prevenção e repressão ao crime, realizassem ações que trouxessem resultados práticos para a sociedade brasileira. Várias foram as contribuições: projetos normativos, iniciativas legislativas, ações estruturantes e de capacitação, compartilhamento de conhecimento, criação de banco de dados e sistemas, e até o aprimoramento de ferramentas de trabalho às instituições competentes (Ministério da Justiça e Segurança Pública, 2022).

Podemos citar o exemplo da Lei nº 12.850/2013[2], que define organização criminosa e dispõe sobre investigação criminal, que teve origem na ENCCLA, trazendo maior segurança jurídica e efetividade aos trabalhos de investigação policial (Brasil, 2013). Também podemos mencionar o Simba, Sistema de Investigação de Movimentações Bancárias, uma referência entre as instituições policiais e ministérios públicos, como uma ferramenta de trabalho muito importante que foi objeto de ação da ENCCLA, e vem potencializando os trabalhos de investigação e análise financeira de inúmeros investigadores que atuam pelo Brasil. Além disso, uma rede de laboratórios de tecnologia foi criada para o enfrentamento da lavagem de dinheiro (Rede-Lab[3]) como resultado da Meta 16/2006 da ENCCLA, após discussões em ação específica, com a finalidade de propiciar o aperfeiçoamento e a evolução tecnológica de diversas instituições de persecução penal (Ministério da Justiça e Segurança Pública, 2022, p. 45).

Dentre os vários resultados obtidos ao longo de anos (Ministério da Justiça e Segurança Pública, 2022, p. 12, grifo nosso), é pertinente mencionar as ações concretas para:

> [...] III. criação e difusão de banco de dados e integração de dados, a exemplo do Sistema Nacional de Bens Apreendidos (SNBA), Sistema de Investigação de Movimentações Bancárias (Simba), Sistema de Fornecimento de Informações ao Poder Judiciário (Infojud), Cadastro Nacional de Clientes do Sistema Financeiro (CCS), Cadastro Nacional de Entidades Sociais (CNES) e Cadastro de Entidades inidôneas e Suspeitas (Ceis), **Cadastro de Pessoas Politicamente Expostas (PEP)**; [...]

[2] https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2013/lei/l12850.htm
[3] https://www.gov.br/mj/pt-br/assuntos/sua-protecao/lavagem-de-dinheiro/lab-ld

Uma das principais medidas adotadas através da ENCCLA é a identificação de pessoas expostas politicamente (PEPs) e padronização de ações para os agentes obrigados, aquelas pessoas físicas e jurídicas que passam a ter obrigações de procedimentos específicos ao lidar com pessoas expostas politicamente. A identificação e o monitoramento de PEPs são não apenas justificados, mas necessários no combate à corrupção e à lavagem de dinheiro. Apesar de todo o esforço das forças de segurança pública e demais órgãos colaboradores, ainda existem muitas lacunas referentes às informações não disponíveis para os órgãos de segurança pública e persecução criminal.

Uma dessas lacunas são os relacionamentos entre pessoas investigadas ou monitoradas e criminosos ou "laranjas" (aqui chamados de associação ou vínculo entre entidades), como no caso das pessoas expostas politicamente, no âmbito da estratégia nacional de combate à corrupção e à lavagem de dinheiro. Esses relacionamentos são difíceis de serem identificados e, mais ainda, não são coletados e armazenados pelos órgãos de segurança pública, exceto em casos específicos e durante investigações em andamento. O trecho da matéria jornalística a seguir é um exemplo de conteúdo de fonte aberta no qual podemos ver a associação de um PEP com outras entidades e relacionamentos (pessoa, organização, local, e "foi preso"):

> O envolvimento de políticos com o tráfico não é novidade. Em 1991, o então deputado federal **Jabes Rabelo**, cuja família construiu um império financeiro de origem suspeita em **Cacoal**, **Rondônia**, teve o mandato cassado pela **Câmara** depois que seu irmão, **Abidiel Rabelo**, **foi preso** em **São Paulo** com 554 quilos de cocaína portando uma falsa carteira de assessor parlamentar com a assinatura do irmão. (Quadros, 2020) (grifo nosso)

Neste contexto, a implementação de ferramentas de mineração de texto e extração de relacionamentos ajustadas para o domínio político brasileiro em idioma português do Brasil seria de grande utilidade para investigações e levantamentos preliminares de pessoas suspeitas ou envolvidas em crimes diversos, em fontes abertas (*Web*), especialmente àqueles relacionados à corrupção e lavagem de dinheiro.

2.4 ALINHAMENTO COM A CIÊNCIA DA INFORMAÇÃO

A Ciência da Informação (CI) é uma área que ocupa cada vez mais destaque no cenário científico, especialmente por tratar de um objeto que é compartilhado com várias

outras áreas, qual seja, a informação. Ela investiga as propriedades e o comportamento da informação. Uma ciência interdisciplinar derivada e relacionada a várias outras como matemática, lógica, linguística, psicologia, ciência da computação, comunicação, biblioteconomia, para citar algumas. A CI realiza pesquisas sobre o processamento de informação para melhorar a acessibilidade e usabilidade da informação. Isso inclui estudo da representação da informação em sistemas naturais e artificiais, assim como códigos para sua transmissão eficiente e o estudo de técnicas e dispositivos para processamento de informação (Borko, 1968).

O uso das mídias sociais expandiu o poder de geração de conteúdo também para as pessoas e aumentou significativamente a quantidade de material disponível para consumo no ambiente online. Como ressalta Castells (2005, p. 69), "pela primeira vez na história, a mente humana é uma força direta de produção, não apenas um elemento decisivo no sistema produtivo". Na década de 1990, Saracevic (1996) pontuou um importante paralelo, entre as questões científicas, à qual a Ciência da Informação se dedica a estudar, assim como às questões práticas (profissionais) com as quais lida. Esses problemas envolvem a efetiva comunicação do conhecimento e seus registros entre os seres humanos, no contexto social, institucional e individual do uso e das necessidades da informação.

Assim sendo, as concepções de informação-como-coisa e informação-como-processo descritos por Buckland (1991) no âmbito da CI, estabelecem uma base sólida para compreender a dinâmica das atividade de inteligência e de investigação policial, as quais se baseiam na busca do dado negado (inteligência clássica) e na identificação de autoria e materialidade do crime (persecução criminal). A Ciência da Informação continua a desempenhar um papel fundamental na compreensão e na gestão da informação em um mundo cada vez mais conectado e digitalizado, onde as organizações investem massivamente na prospecção de dados como uma ferramenta essencial para aprimorar o processo de tomada de decisões. Isso é reflexo da crescente compreensão do valor dos dados na era digital, conforme destacado por Aular e Pereira (2007) e Islam (2018). Sua natureza interdisciplinar permite abordar os desafios complexos relacionados à produção, distribuição, acesso e uso da informação em uma variedade de contextos, desde a academia até as práticas profissionais. Em meio às rápidas mudanças no cenário

tecnológico e social, a CI continua a se manter relevante, adaptando-se e expandindo-se para enfrentar os novos desafios e oportunidades que surgem no campo da informação.

2.5 DELIMITAÇÃO DA PROPOSTA

Tendo em vista a especificidade do domínio político brasileiro e o foco deste trabalho, que é a análise da extração de relacionamentos por meio de técnica específica, qual seja, *Natural Language Inference*, optou-se por manter seu escopo limitado para que fosse exequível no cronograma proposto.

Desta forma, uma vez que um dos procedimentos consistiu na criação de um *corpus* textual de apoio a partir de coleta de documentos na *Web*, foi necessário delimitar as palavras-chaves utilizadas para busca, escolhidas com base nas tipificações penais que mantêm relação com os crimes cometidos por pessoas expostas politicamente. Este trabalho não abrange a totalidade das palavras relacionadas às tipificações, mas, de forma geral, procurou-se abranger uma quantidade de palavras que são significativas no contexto com base nas tipificações penais. Além disso, outro critério definido para a pesquisa é que somente o corpo do texto das notícias fosse coletado: sem títulos, códigos de programação e eventuais propagandas apresentadas nas páginas *HTML* originais. A proposta também definiu um escopo reduzido com relação às entidades a serem trabalhadas, tanto na construção do corpus quanto na extração de relacionamentos. Optou-se por utilizar exclusivamente as seguintes entidades nomeadas: Pessoa, Organização, Tempo e Local.

Importante frisar, ainda, que seria impossível obter uma infinidade de conteúdos de websites diversos, com grande variação temática para este trabalho. Também foi considerado que seria pouco relevante a origem dos documentos extraídos da *Web*, ou seja, de qual empresa ou site coletou-se cada reportagem utilizada. O cerne do trabalho foi a obtenção de textos sobre os assuntos propostos, com entidades, para que fosse possível trabalhá-los diante do cenário de aplicação definido. Não foi avaliado se determinado website ou empresa foi mais ou menos acessado, ou se por questão comercial (acesso restrito) alguma outra deixou de ser consultada.

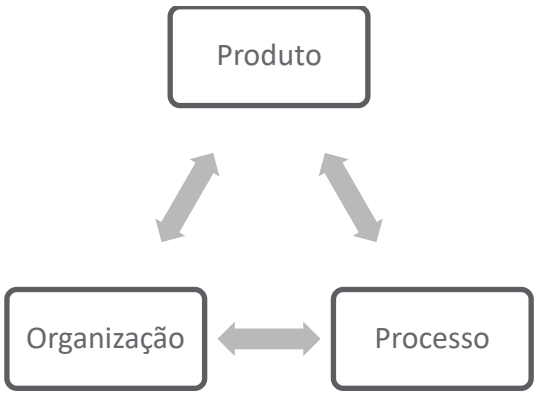
2.6 ESTRUTURA

Este trabalho está dividido em nove capítulos. O primeiro capítulo contém a introdução. No segundo capítulo, apresentamos os objetivos, a justificativa, o alinhamento com a Ciência da Informação, a delimitação da proposta e a estrutura do trabalho. No terceiro capítulo, apresentamos uma revisão sobre a informação na atividade de inteligência e na investigação policial. No quarto capítulo, abordamos uma revisão sobre Pessoas Expostas Politicamente. No quinto capítulo, discorremos sobre o Processamento de Linguagem Natural, os Transformers, BERT, Reconhecimento de Entidades Nomeadas e Extração de Relacionamentos, além das principais técnicas e métricas utilizadas na avaliação de resultados. No sexto capítulo, apresentamos a Metodologia, com a caracterização da pesquisa, os procedimentos metodológicos, a revisão sistemática da literatura e os trabalhos relacionados. No sétimo capítulo, apresentamos os elementos de suporte da proposta. No capítulo oito, apresentamos a proposta de pipeline, com a construção do corpus textual e a proposição das verbalizações e toda a sua execução. No nono capítulo, apresentamos as considerações finais e trabalhos futuros.

3 INFORMAÇÃO NA ATIVIDADE DE INTELIGÊNCIA E NA INVESTIGAÇÃO POLICIAL

No final da década de 1940, o professor norte-americano Sherman Kent deixou importantes contribuições para a atividade de inteligência através do seu livro *Strategic Information*, que abordou a "Informação" sobre três vertentes: conhecimento, organização e atividade, que também podem ser entendidos como produto, organização e processo (Kent, 1967). A primeira vertente aborda informações propriamente dita como uma espécie de conhecimento (produto), destinado a assessorar o processo decisório. A segunda aborda a organização e administração das informações, como os serviços secretos que atuam na busca do dado negado[4] e na produção do conhecimento de inteligência. A terceira, por sua vez, é voltada para as informações como atividade desempenhada por uma organização. Refere-se aos processos desenvolvidos para a obtenção de determinados dados, informações e inteligência. Esse modelo é descrito por tridimensional e é representado na Figura 1.

Figura 1 - Definição tridimensional do conceito de Inteligência.



Fonte: elaborado pelo autor.

Conforme definição de Gonçalves (2010, p. 12–16), informação é gênero e inteligência é espécie. Seguindo essa taxonomia, toda inteligência é informação, mas nem toda informação é inteligência. Gonçalves (2010), ressalta ainda, que se não houver

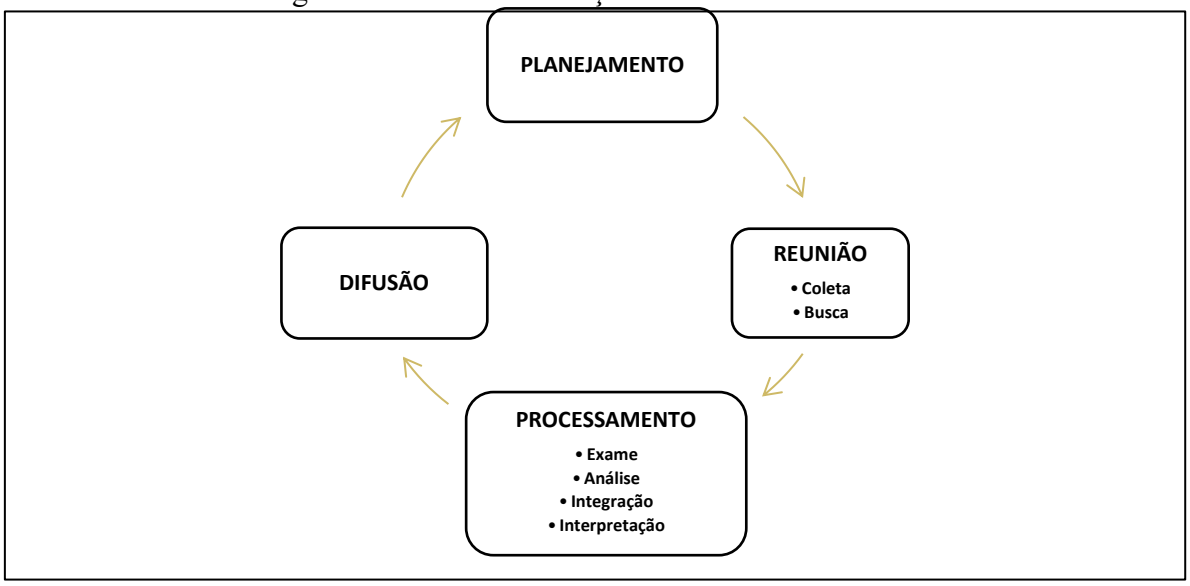
[4] Busca do dado negado é um jargão da Atividade Inteligência que se refere a uma ação que utiliza técnicas sigilosas para obter dado de interesse do Órgão de Inteligência que esteja sendo protegido pela pessoa/órgão/Estado que o detém. É atividade típica de Serviços Secretos. (Gonçalves, 2010, p. 11–12; Schauffert; Lento, 2011).

componentes de sigilo em suas fontes ou caráter sigiloso em seu conteúdo, o produto não pode ser assinalado como inteligência. Esse caráter de sigilo pode ser não necessariamente pelos dados nele utilizados, mas pela análise realizada.

Embora a expressão mais comum no Brasil seja Inteligência Policial, fato é que o nome Inteligência Criminal é mais aceito academicamente, já que não só as polícias atuam nessa atividade, mas também outras instituições, como Ministério Público, o Exército, o Sistema Prisional, a Guarda Municipal etc. Também, não se deve nomear uma atividade de pesquisa pelo nome do pesquisador, e sim pelo objeto de pesquisa. Assim sendo, a Inteligência Criminal é uma atividade especializada e detentora de técnicas e métodos próprios (Mingardi, 2007).

Uma definição ampla de Inteligência nos conduz para um conhecimento específico, fruto de uma informação analisada, que pode servir para um propósito específico (tomada de decisão). Na Figura 2, apresentamos o Ciclo de Produção do Conhecimento, que caracteriza as etapas relacionadas ao processo pelo qual a informação é reunida, convertida em inteligência e disponibilizada aos consumidores, que são os tomadores de decisão. Existe certa variação nas fases do processo, que pode chegar a dez fases, incluindo etapas prévias de planejamento e requisitos, bem como etapas posteriores de consumo e avaliação, na visão de diferentes autores (Baierle et al., 2011; Barbosa, 2012; Brasil, 2014; Cepik, 2003a; Fernandes, 2006; Mingardi, 2007).

Figura 2 - Ciclo de Produção do Conhecimento.



Fonte: Adaptado de Gonçalves (2010, p. 69).

No caso específico da Inteligência Criminal, o objetivo final é subsidiar o processo decisório no âmbito da Segurança Pública. Vejamos o conceito de Inteligência de Segurança Pública ou Inteligência Criminal, encontrado na Resolução nº 1, de 15 de julho de 2009, que regulamenta o Subsistema de Inteligência de Segurança Pública (SISP) [5]:

> É a atividade permanente e sistemática, via ações especializadas, que visa identificar, acompanhar e avaliar ameaças, reais ou potenciais, sobre a segurança pública e **produzir conhecimentos e informações** que subsidiem o planejamento e a execução de políticas de Segurança Pública, bem como ações para prevenir, neutralizar e reprimir atos criminosos de qualquer natureza, de forma integrada e em subsídio à investigação e à produção de conhecimentos. (Brasil, 2009) (grifo nosso)

Em relação à Inteligência Policial, tratou a resolução supracitada de também defini-la:

> É o conjunto de ações que empregam técnicas especiais de investigação, visando a confirmar evidências, indícios e a obter conhecimentos sobre a atuação criminosa dissimulada e complexa, bem como a identificação de redes e organizações que atuem no crime, de forma a proporcionar um perfeito entendimento sobre a maneira de agir e operar, ramificações, tendências e alcance de condutas criminosas. (Brasil, 2009).

A atividade de Inteligência pode ser classificada, segundo seu escopo, em uma diversidade de categorias. A Inteligência Militar ou de Estado é a mais antiga e deu origem aos serviços de inteligência da atualidade. Existe, contudo, um escopo muito diferenciado entre a inteligência militar e a inteligência civil. Atualmente, muitos países possuem sistemas de inteligência estruturados que contam além dos serviços de inteligência civil, também com agências e órgãos de inteligência militar, vinculados a cada força armada e/ou Ministério da Defesa. O Quadro 1 apresenta a classificação de Inteligência quanto à sua categoria de acordo com Gonçalves (2010, p. 21–43).

Uma importante matéria relacionada à atividade de Inteligência se refere às fontes e meios de obtenção de dados. As fontes podem ser classificadas quanto à sua confidencialidade ou quanto à sua origem dos dados. Os meios de coleta e as fontes de informação definem disciplinas bastantes especializadas em inteligência, que inclusive a

[5] https://www.normasbrasil.com.br/norma/resolucao-1-2009_111521.html

literatura internacional define acrônimos derivados do uso norte-americano, conforme se vê no Quadro 2.

Quadro 1 - Classificação das Inteligências quanto à sua categoria.

Inteligência Militar	Visa subsidiar o processo decisório dos vários escalões das forças armadas.
Inteligência Policial ou Criminal	Relacionada a questões táticas de repressão e investigação de ilícitos e grupos infratores.
Inteligência Financeira	Ações voltadas à identificação de delitos financeiros, pessoas e organizações relacionadas.
Inteligência Fiscal	Voltada para identificação e investigação de delitos contra a ordem tributária.
Inteligência Competitiva	Voltada ao mundo dos negócios. Tem relação com a competição e sobrevivência das empresas.
Inteligência Estratégica	Produção de conhecimentos estratégicos para assessorar os tomadores de decisão do alto escalão de um governo, geralmente com a formulação de cenários prospectivos.
Inteligência de Estado (Externa e Doméstica)	Associada a segurança do Estado e da sociedade, subsidia o processo decisório da mais alta esfera de governo.

Fonte: Compilado pelo autor baseado em Gonçalves (2010, p. 21–43).

Uma área da Inteligência que lida com dados expostos na Internet é denominada de *Open Sources Intelligence (OSINT*) ou Inteligência de Fontes Abertas/Ostensivas, e é dedicada para a produção de conhecimento a partir de material coletado de bases de dados eletrônicas de dados públicos ou privados, especialmente da *Web* (Cepik, 2003b, p. 51). Sua importância cresceu muito após o início da sociedade da informação com a consequente "explosão informacional", que aconteceu após o final da segunda guerra mundial (Bush, 1945). Entretanto, diante dessa grande quantidade de dados, o grande problema do analista passa a ser identificar o que é relevante, i.e., processar toda essa informação e extrair um conhecimento de inteligência.

Quadro 2 - Classificação das fontes.

Confidencialidade	Fontes abertas	OSINT
	Fontes classificadas (dado negado)	-
Origem dos dados	Fontes humanas	HUMINT
	Fontes técnicas	TECHINT

Fonte: Elaborado pelo autor baseado em Gonçalves (2010, p. 78).

Outro fato interessante é a mudança nas práticas de inteligência causado pela grande quantidade de dados de OSINT. Por exemplo, nos dias de hoje, um dos primeiros passos na coleta de dados para produção de conhecimento é a pesquisa por fontes abertas em buscadores da Internet, em busca de dados, imagens, contatos, croquis, mapas etc.

As fontes humanas (HUMINT) eram, antes da "sociedade da informação", as mais tradicionais e mais baratas formas de reunião de dados pela Inteligência. São obtidas a partir de pessoas, inclusive com espionagem na busca do dado negado (fontes classificadas). O trabalho do espião é justamente obter, por qualquer meio, as informações (dado negado) que um inimigo ou concorrente gostaria de manter em segredo. As fontes humanas podem ser oficiais ou não-oficiais, orgânicas (aquelas que pertencem a um serviço de inteligência) ou não-orgânica (qualquer pessoa "recrutada", denominada "agente").

A inteligência técnica ou tecnológica envolve uma série de subcategorias como inteligência de sinais (SIGINT), inteligência fotográfica (PHOTINT), inteligência de imagens (IMINT), inteligência de comunicações (COMINT), inteligência eletrônica (ELINT), inteligência telemétrica (TELINT) e até a inteligência relacionada à interpretação de ondas e sinais eletromagnéticos ou "assinaturas físicas", denominada MASINT. Nos Estados Unidos da América, a SIGINT é tão relevante que durante décadas o maior orçamento daquele governo foi destinado para a agência encarregada de inteligência de sinais naquele país, a *National Security Agency (NSA)*, conforme nos revela Gonçalves (2010, p. 86).

A investigação policial é uma atividade que atua diretamente nas atividades de Polícia Judiciária, na elucidação de crimes, visando a identificação de autoria e materialidade, além das circunstâncias do fato típico. A Constituição Federal de 1988[6] define no art. 144 as competências das polícias civis dos Estados e da União para apurações de infrações penais. Toda informação produzida através dos trabalhos de investigação policial precisa se assegurar da idoneidade da cadeia de custódia. Isso significa que é preciso registrar todo o processo de produção de provas: origem, meios de acesso etc. As técnicas utilizadas para investigação são bem definidas e legalmente amparadas. Diferente da atividade de Inteligência que, apesar de fiscalizada pelo Senado Federal, sua produção de conhecimento não tem o ônus de identificar origens, meios de acesso ou fontes.

A revisão apresentada destaca a importância da inteligência no contexto político, especialmente no campo da segurança pública e investigação criminal. Ambas as

[6] https://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm

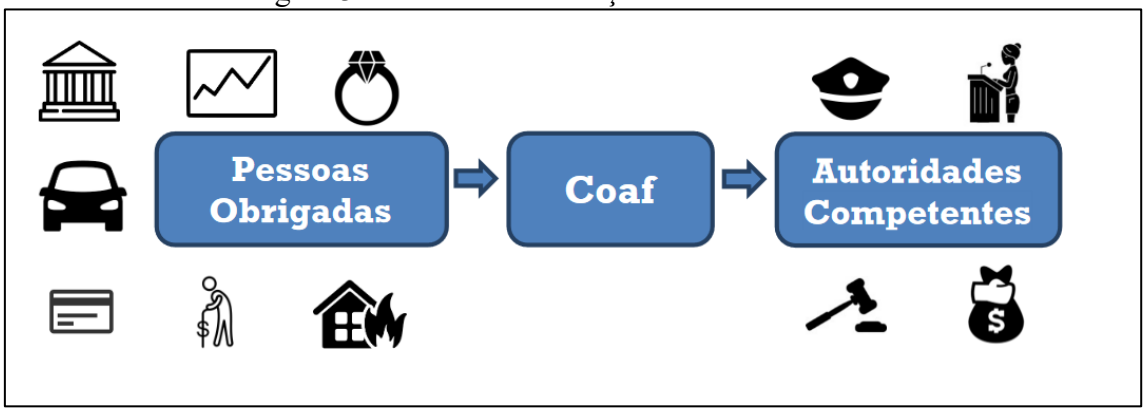
atividades são fundamentais para garantir a ordem e a aplicação da lei em uma sociedade. A análise de dados e informações, especialmente aquelas derivadas do processamento de linguagem natural em textos políticos, desempenha um papel crucial na geração de conhecimento útil para orientar estratégias e ações nessas áreas. É por meio dessa análise que se obtém insumos valiosos para a prevenção e investigação de crimes, bem como para o desenvolvimento de políticas de segurança eficazes.

4 PESSOAS EXPOSTAS POLITICAMENTE (PEPs)

Em 1998, foi publicada a Lei nº 9.613 (Brasil, 1998), tipificando crimes de lavagem e ocultação de bens. Por esse mesmo dispositivo foi criado o Conselho de Controle de Atividades Financeiras (COAF[7]), um importante órgão de fiscalização e controle no âmbito do Ministério da Fazenda. Por meio dessa lei, surge a figura da pessoa obrigada (Figura 3): pessoas físicas e jurídicas que passam a ter determinações legais para adoção de procedimentos especiais de prevenção e combate ao crime de lavagem de dinheiro e financiamento ao terrorismo. Por exemplo, agentes financeiros (bancos) passam a ter o dever de identificar clientes, manter seus registros atualizados e comunicar determinados tipos de operações financeiras ao Banco Central do Brasil, seu agente regulador.

Assim como as instituições financeiras, várias outras pessoas também passam a ter suas atividades controladas pelos agentes reguladores, tais como operadoras de planos de assistência à saúde, comércio de joias, pedras e metais preciosos, comércio de bens de luxo e alto valor, fomento comercial (*factoring*), atividades de compra e venda de joias, cartórios e registradores, juntas comerciais, empresas de transporte e guarda de valores, comércio de antiguidades e/ou obras de arte de qualquer natureza dentre outros (COAF, 2020).

Figura 3 - Fluxo de informação através do COAF.



Fonte: (COAF, 2022, p. 7)

Em 2003, com a criação de um órgão colegiado denominado Estratégia Nacional de Combate à Corrupção e à Lavagem de Dinheiro (ENCCLA), uma série de ações

[7] https://www.gov.br/coaf/pt-br

passam a ser tomadas como o propósito de aprimorar a estrutura brasileira de combate à corrupção, à lavagem de dinheiro, ao financiamento do terrorismo e ao crime organizado (Ministério da Justiça e Segurança Pública, 2022, p. 5). Na edição de 2013, como resultado da ação nº 07 da ENCCLA (Quadro 3), foi criado o cadastro de Pessoas Expostas Politicamente (PEPs).

Quadro 3 - Ação 7 da ENCCLA, edição 2013.

AÇÃO 7	Implementar cadastro de Pessoas Expostas Politicamente (PEPs) com acesso público.
Coordenadores	CGU e COAF
Colaboradores	BACEN, CNJ, CNMP, CVM, FEBRABAN, MPF, RFB, SUSEP, FPCC/RS, Câmara dos Deputados, STF e TCU. (Convidar PREVIC, MPOG/SEGEP, Casa Civil, Senado e Imprensa Nacional).

Fonte: ENCCLA (2013).

Por meio desse cadastro, ficam disponíveis os dados, em formato aberto, de agentes públicos que desempenham ou tenham desempenhado, nos últimos cinco anos, cargos, empregos ou funções públicas relevantes. A Controladoria-Geral da União (CGU) organizou e mantém atualizado o conjunto de dados a partir de informações disponibilizadas por vários setores e entidades da Administração Pública. O cadastro contém a identificação de titulares de cargos e de funções públicas listas na regulamentação específica como indicadores da condição de PEP (Brasil, 2021b).

A Resolução COAF nº 40, de 22 de novembro de 2021, atualizou os procedimentos a serem observados em relação a pessoas expostas politicamente, por aqueles que se sujeitam à supervisão do COAF (BRASIL, 2021c). Por meio desse dispositivo, também é feito um rol taxativo das pessoas consideradas PEPs, que pode ser encontrado no ANEXO A. Para efeitos ilustrativos mencionamos aqui os principais: detentores de mandatos eletivos dos Poderes Executivos e Legislativos da União, Estados e Municípios, Ministros de Estado ou equiparado, Presidente, Vice-Presidente e Diretor de entidades da administração pública indireta, cargos com DAS[8] de nível 6 ou equivalente, membros de Conselhos e Tribunais superiores dentre outros, além dos seus familiares, estreitos colaboradores e ou pessoas jurídicas de que participam.

[8] Direção e Assessoramento Superior - DAS

Como estreito colaborador, a resolução supracitada esclarece que pode ser pessoa com qualquer outro tipo de estreita relação de conhecimento público com uma pessoa exposta politicamente, além dos outros mencionados no inciso I, do § 2º, artigo 2º, da mencionada resolução (grifo nosso):

> I - pessoas naturais que são conhecidas por terem sociedade ou propriedade conjunta em pessoas jurídicas de direito privado ou em arranjos sem personalidade jurídica, que figurem como mandatárias, ainda que por instrumento particular, ou possuam **qualquer outro tipo de estreita relação de conhecimento público** com uma pessoa exposta politicamente;

Na definição de organização criminosa, Lei nº 12.850[9], de 02 de agosto de 2013, existe a previsão, no artigo 2º caput, de utilização de pessoa interposta (laranja ou testa de ferro) na tipificação de promover, constituir, financiar ou integrar organização criminosa. Desta forma, o ordenamento jurídico tem a preocupação de não deixar escapar, através de brecha, por falta de tipificação, a previsão legal de cometimento de crime através de pessoas associadas (testas de ferro ou laranjas).

Este cenário, que envolve legislação e a abundância de informações em fontes abertas apresenta um ambiente propício para aplicação da área de Ciência da Informação, com o uso de tecnologias computacionais para extração de informações. Neste contexto, torna-se essencial a revisão da área de Processamento de Linguagem Natural para o desenvolvimento e análise dessas informações.

[9] https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2013/lei/l12850.htm

5 PROCESSAMENTO DE LINGUAGEM NATURAL

Desde a década de 1950, iniciaram-se os primeiros esforços para o processamento de linguagem natural. Com o surgimento dos primeiros computadores digitais, os pesquisadores buscavam maneiras de permitir que as máquinas compreendessem e manipulassem a linguagem humana, ou seja, a forma que os humanos falam e escrevem. Existe um certo consenso de que um memorando de *Warren Weaver*, que era diretor da Divisão de Ciências Naturais da Fundação Rockefeller, em julho de 1949, trouxe a ideia de utilizar os computadores digitais recentemente inventados para traduzir documentos entre uma linguagem natural e outra. Ele sugeriu usar ideias da criptografia e da teoria da informação para tradução de idiomas, antes mesmo que a maioria das pessoas tivessem qualquer ideia do que os computadores poderiam ser capazes. O trabalho com o PLN continua até os dias atuais e enfrenta diversas dificuldades, pois a comunicação humana é repleta de nuances e complexidades (Hutchins, 1999).

Pela definição de Liddy (2001, p. 1), o Processamento de Linguagem Natural engloba uma variedade de técnicas computacionais voltadas para a análise e representação de textos que ocorrem de forma natural em diferentes níveis de análise linguística. O objetivo principal é alcançar um processamento de linguagem que se assemelhe ao humano, a fim de viabilizar uma série de tarefas ou aplicações. Apesar de sua denominação, o campo do processamento de linguagem natural compreende duas abordagens distintas: o processamento de linguagem e a linguagem generativa. O primeiro foco refere-se à análise da linguagem com o propósito de produzir uma representação coerente e significante, enquanto o segundo visa gerar linguagem com base em uma representação prévia. A tarefa de processamento assemelha-se à função de um leitor ou ouvinte, enquanto a tarefa de geração é comparável à de um escritor ou orador.

O processamento de linguagem natural fornece teoria e implementação para uma variedade de aplicações. Qualquer aplicação que envolva o uso de texto é candidata a utilizar o PLN. As aplicações mais comuns estão listadas no Quadro 4. Muitas aplicações de PLN possuem como tarefa básica o reconhecimento de entidades nomeadas (REN), que visa a identificação de entidades com significado específico no texto, tais como nomes de pessoas, nomes de lugares, nomes de organizações, nomes próprios, dentre outras. Ela é parte indispensável de muitos métodos de PLN, como extração de

informações (EI), recuperação de informações (RI), tradução automática e sistemas de respostas a perguntas.

Quadro 4 - Aplicações mais comuns em PLN.

Recuperação da Informação (RI)	Recuperação de informações relevantes de grandes conjuntos de dados, geralmente documentos de textos, em resposta a uma consulta ou necessidade do usuário.
Extração de Informação (EI)	Reconhecimento, marcação e extração em uma representação estruturada de certos elementos-chave de informações em grandes coleções de textos, como pessoas, empresas, locais, organizações.
Sumarização de textos	Reduz um texto maior a uma representação narrativa abreviada mais curta, ricamente constituída do documento original.
Tradução automática	Tradução de textos entre idiomas
Sistemas de diálogo	Sistemas que realizam interações mais naturais e fluídas com os usuários de forma bidirecional.
Geração de texto	Criação automática de textos com base em modelos e dados de entrada (sistemas de diálogos, Perguntas e Respostas)
Análise de sentimentos	Avaliação e classificação do sentimento expresso em textos.

Fonte: elaborado pelo autor.

Um ponto importante para a compreensão do que seja o processamento de linguagem natural é a noção do nível de análise linguística. Isso reside no fato de que existem vários tipos de processamento de linguagem atuando quando os humanos produzem ou compreendem a linguagem. Acredita-se que os humanos normalmente utilizam todos estes níveis, já que cada nível transmite diferentes tipos de significado. Os sistemas de PLN utilizam diferentes níveis ou combinações de níveis de análise linguística, o que causa certa confusão por parte dos não especialistas sobre o que realmente é a PLN (Liddy, 2001, p. 2).

No Quadro 5, apresentamos os níveis de representação e processamento linguístico em ordem de complexidade conforme sentido da seta. O nível de representação mais básico é o fonológico, que trata de sons de fala dentro e entre palavras. Existem três tipos de regras para análise fonológica: regras fonéticas, relacionadas aos sons dentro de palavras; regras fonêmicas, para variações de pronúncia quando as palavras são faladas juntas; e regras prosódicas, para variações de acento e entonação ao longo de uma frase. Em um sistema de PLN que tem entrada por áudio, os sons são analisados e codificados em sinal digital para interpretação por regras ou modelos de linguagem específico (Liddy, 2001, p. 6).

O nível morfológico trata dos componentes das palavras, os morfemas. Uma decomposição morfológica pode resultar em vários morfemas distintos, tais como prefixos, sufixos e raiz morfológica. É interessante observar que o significado de cada

morfema é o mesmo, o que permite que os humanos compreendam uma nova palavra pela sua decomposição em morfemas (Pardo, 2023). No nível lexical acontece a interpretação do significado das palavras individualmente, atribuindo a elas uma única etiqueta de classe gramatical. Palavras com mais de um significado recebem uma *tag* (etiqueta ou anotação) da classe gramatical mais provável. Esse nível pode exigir um léxico que varia em complexidade, desde um léxico simples, contendo apenas palavras e suas partes do discurso, até léxicos mais complexos, contendo informações sobre a classe semântica da palavra, seus argumentos e as restrições sobre esses argumentos (Liddy, 2001).

Quadro 5 - Níveis de representação e processamento linguístico.

	**Pragmático**	Significado além do texto (intenções, planos e objetivos)
	Discursivo	Propriedades do texto
	Semântico	Significado
	Sintático	Análise das palavras de uma frase
	Lexical	Significado individual das palavras
	Morfológico	Formação das palavras
	Fonológico	Identificação dos sons

Fonte: Compilado pelo autor baseado em Pardo (2023) e Liddy (2001).

Em relação ao nível sintático, este se concentra na análise das palavras de uma frase e como elas se combinam. Essa análise revela as relações de dependência estrutural entre as palavras. A sintaxe é fundamental para a transmissão de significado na maioria das línguas, pois a ordem e a dependência das palavras contribuem para o significado. Nos níveis Discurso e Pragmático temos os estágios mais complexos de análise linguística. O nível Discurso lida com unidades de texto maiores do que uma simples frase. Essa análise concentra-se nas propriedades do texto como um todo, revelando significado ao estabelecer conexões entre sentenças que compõem o texto. Por outro lado, o nível Pragmático vai além do texto, utilizando o contexto além do conteúdo do texto para compreensão. Ele engloba a compreensão de intenções, planos e objetivos subjacentes. Um significado que é lido nos textos sem realmente estar codificado nele. Por exemplo, a geração de texto de um subordinado para um chefe pode diferir da geração de texto do chefe para o subordinado, refletindo diferentes perspectivas (Pardo, 2023).

Uma etapa fundamental do PLN é a tokenização, que consiste em dividir o texto em *tokens* separados por espaços em branco, incluindo espaços, tabulações e quebras de linhas. Dependendo da aplicação, alguns *tokens* especiais podem ser adicionados, como

tokens de início e fim de sentença, tokens desconhecidos para palavras não encontradas no vocabulário etc. Cada *token* pode até ser dividido em subpalavras ou até mesmo caracteres individuais para capturar melhor informações semânticas e morfológicas. Após a tokenização, o texto é representado como uma sequência de *tokens* e pode facilmente ser processado por modelos de PLN, como redes neurais e algoritmos de aprendizado de máquina (Souza; Nogueira; Lotufo, 2019, p. 4–5).

5.2 TRANSFORMERS

Em 2017, o *Google Brain Team* apresentou um trabalho intitulado *Attention Is All You Need* (Vaswani et al., 2017), com uma arquitetura baseada em transformadores, que evita a recorrência e, em vez disso, depende inteiramente de mecanismos de atenção para desenhar dependências globais entre entrada e saída. Desde então, esta arquitetura tem influenciado fortemente os modelos propostos nos anos seguintes. O mecanismo de *self-attention* (autoatenção) recebe n entradas e retorna n saídas. O que acontece, basicamente, é que esse mecanismo permite que as entradas interajam entre si, por isso o termo auto (*self*), e descubram a quem devem prestar mais atenção. As saídas são resultado dessas interações e pontuações de atenção. Assim, é possível fazer paralelização, reduzindo o tempo de treinamento.

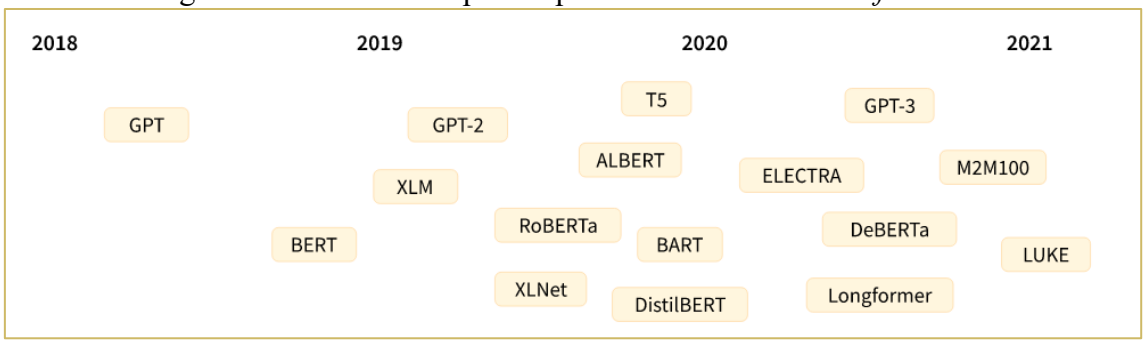
5.2.1 Bidirecional *Encoder Representations From Transformers* (BERT)

Um modelo de representação de linguagem chamado BERT, que significa bidirecional *encoder representations from transformers*, foi introduzido pelo *Google AI Language* em 2019 (Devlin et al., 2019). Ele foi projetado para pré-treinar representações bidirecionais profundas a partir de texto não rotulado, considerando simultaneamente o contexto à esquerda e à direita em todas as camadas, por meio de cabeças de autoatenção. Uma cabeça de autoatenção refere-se a uma unidade dentro de um mecanismo de atenção usado no modelo para calcular as associações entre todas as palavras em uma sequência de entrada. Ao ter várias cabeças de autoatenção, o modelo é capaz de aprender representações mais ricas e detalhadas entre as palavras, o que contribui para uma melhor compreensão e geração de texto em tarefas de PLN. A menor arquitetura desde modelo

conta com 12 camadas de codificadores (blocos transformadores), uma camada oculta de tamanho 768 e 12 cabeças de autoatenção, sendo 110 milhões de parâmetros nessa versão básica, denominada BERT Base.

Esta arquitetura substituiu parcialmente arquiteturas tradicionais para resolução de problemas de linguagens como *Long Short-Term Memory* - LSTM (Devlin et al., 2019). A LSTM é uma arquitetura baseada em redes neurais recorrentes (RNN) que processa o texto sequencialmente. Embora a LSTM tenha a capacidade de aprender e reter padrões de longo prazo, evitando o problema do desvanecimento do gradiente, que afeta muitas RNNs tradicionais, ela requer treinamento específico para tarefas de PLN, com um grande volume de dados rotulados, o que consome tempo significativo. Na Figura 4, observamos a evolução dos primeiros modelos de *Transformers*, com a criação de muitas variações do modelo BERT, como RoBERTa, ALBERT, DeBERTa, entre outros, apresentadas na linha de tempo.

Figura 4 – Linha do tempo dos primeiros modelos *Transformers*.



Fonte: https://huggingface.co/learn/nlp-course/pt/chapter1/4

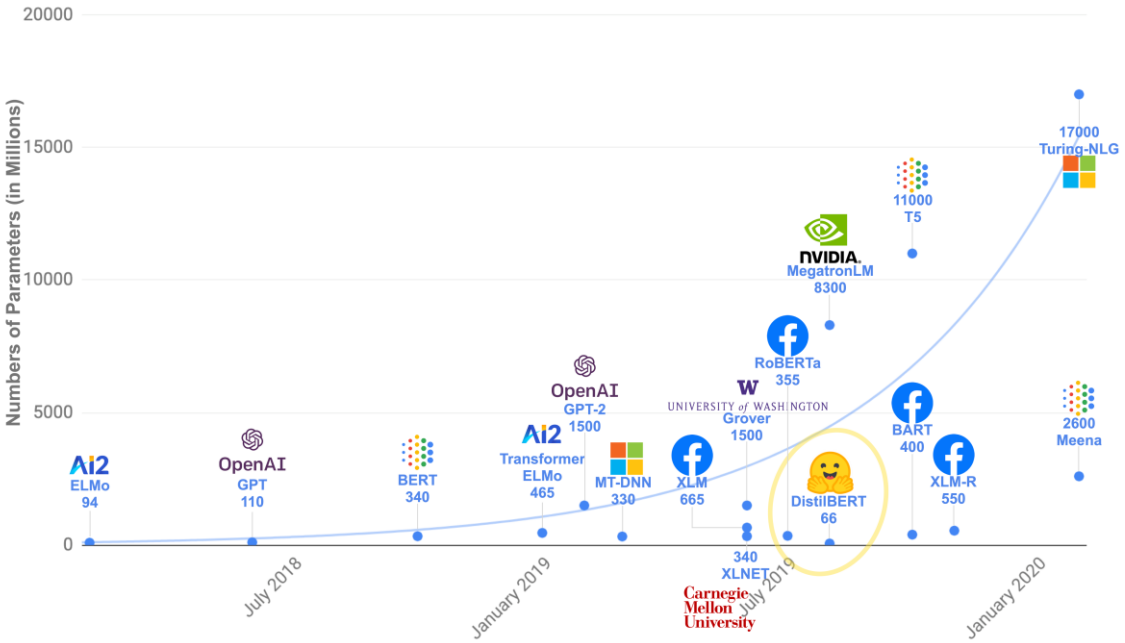
Ainda segundo Devlin et al. (2019), o modelo BERT Base foi dimensionado para ter o mesmo tamanho do OpenAI GPT[10], permitindo uma comparação direta entre os dois. No entanto, também existe uma versão expandida do BERT, conhecida como BERT Large, que possui 24 camadas de codificadores, uma camada oculta de tamanho 1024 e 16 cabeças de autoatenção, totalizando 340 milhões de parâmetros. Notavelmente, os modelos maiores apresentam um desempenho superior, mesmo quando treinados em conjunto de dados menores. A Figura 5 oferece uma representação visual da quantidade de parâmetros utilizados nos principais modelos, incluindo BERT e OpenAI.

[10] O OpenAI GPT executa várias tarefas de processamento de linguagem natural, tais como tradução da linguagem natural para códigos, criação de imagens com AI, dentre outros (https://openai.com/).

O ajuste fino (*fine-tuning*) é uma estratégia fundamental no contexto de modelos pré-treinados, como o BERT. Esse processo envolve adaptar um modelo pré-treinado para uma tarefa específica, sem a necessidade de modificar substancialmente sua arquitetura. O BERT, por exemplo, pode ser ajustado com alta performance com apenas uma camada de saída adicional para diversas tarefas, como Perguntas e Respostas (Q&A) e inferência de linguagem. Isso permite uma transferência de aprendizado eficiente, aproveitando o conhecimento prévio do modelo. No entanto, construir um modelo “do zero”, i.e., a partir do estágio inicial, demanda tentativas e erros sucessivos para determinar o número e o tipo de camadas a serem utilizadas, bem como em que ordem as colocar e quantos nós incluir em cada camada (Devlin et al., 2019).

Durante o ajuste fino, uma rede de neurônios pré-treinada é adaptada para uma nova tarefa, eliminando camadas de classificação e de saída e adicionando novas camadas conforme necessário. As camadas da rede pré-treinada podem ser "congeladas" ou treinadas, dependendo do contexto da nova tarefa. Entretanto, o ajuste fino de um modelo, que envolve a atualização de todas as camadas treináveis de uma rede neural pré-treinada usando novos dados, geralmente é mais demorado que o aprendizado por transferência, onde apenas algumas camadas finais são ajustadas.

Figura 5 - Quantidade de parâmetros utilizados nos primeiros modelos ao longo do tempo.



Fonte: https://huggingface.co/learn/nlp-course/pt/chapter1/4

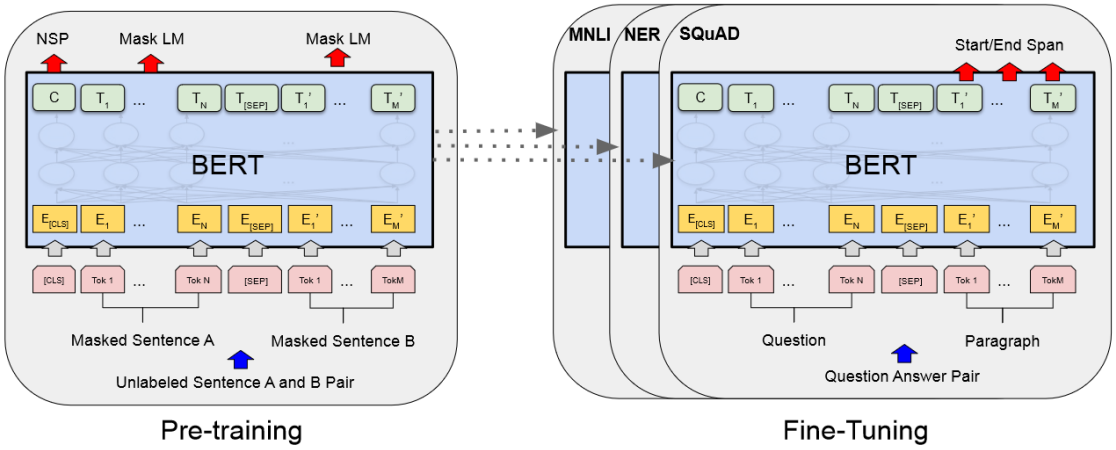
BERT usa a tokenização *WordPiece*[11] para dividir as sentenças em tokens (palavras), com um vocabulário de 30.000 tokens. O trabalho de pré-treino é feito através de duas tarefas não-supervisionadas: *Masked Language Model* (MLM) e *Next Sentence Prediction* (NSP), conforme apresentado na Figura 6. A subtarefa de modelo de linguagem mascarado (MLM) oculta aleatoriamente alguns dos *tokens* da entrada, processo denominado "mascaramento", objetivando fazer a predição da palavra mascarada baseado no seu contexto. O BERT consegue demonstrar a importância do pré-treinamento bidirecional para representação de linguagens, usando o MLM para entender o contexto de uma frase. Essas representações reduzem a necessidade de muitas arquiteturas de tarefas específicas altamente projetadas. O BERT é o primeiro modelo de representação baseado em ajuste fino que alcança alto desempenho superando muitas arquiteturas específicas de tarefas.

Muitas tarefas importantes de processamento de linguagem natural são baseadas no entendimento da relação entre duas sentenças que não é diretamente capturada pela modelagem de linguagem. A fim de treinar um modelo para compreender o relacionamento entre sentenças, faz-se o pré-treino para a tarefa de predição de próxima sentença (NSP), processo que tem como objetivo prever qual frase vem a seguir a uma frase dada. Isso ajuda o modelo a compreender melhor o contexto da sentença e melhora a precisão das tarefas de processamento de linguagem natural, como classificação textual, tradução de idiomas e extração de entidades.

O *corpus* utilizado no pré-treino do BERT foi o *BooksCorpus* (800M de palavras) e a *Wikipedia* em inglês (2.500M de palavras). Da *Wikipedia*, foram utilizadas somente as passagens de textos, ignorando listas, tabelas e títulos. Devlin et al. (2019) chegaram à conclusão que é fundamental usar corpus em nível de documentos em vez de corpus em nível de frases, a fim de extrair longas sequências contíguas. O ajuste fino é feito para cada tarefa, simplesmente conectando as entradas e saídas específicas da tarefa no BERT e ajustando todos os parâmetros. Seu custo de processamento é relativamente baixo.

[11] WordPiece é um algoritmo de tokenização desenvolvido pelo Google especialmente para treinar o BERT. Tem sido utilizado em outros modelos baseados em BERT. Identifica subpalavras e adiciona um prefixo (##) em cada uma delas que estejam dentro da palavra.

Figura 6 - Procedimentos gerais de pré-treinamento e ajuste fino para BERT. Além das camadas de saída, as mesmas arquiteturas são usadas nas duas etapas.



Fonte: Devlin et al. (2019).

Conforme Souza et al. (2020), essa estratégia de trabalho mudou o cenário de PLN nos últimos anos e alcançou o estado da arte em onze tarefas, inclusive o reconhecimento de entidades nomeadas. Além de melhorar a performance, a transferência de aprendizado reduziu a quantidade de dados rotulados necessária para tarefas de aprendizado supervisionado. Souza et al. (2020) treinaram o modelo BERT para o idioma português do Brasil com o apelido de BERTimbau, usando dados do brWac[12], um *corpus* de *web pages* grande e diversificado. Os pesquisadores fizeram análise de tokenização do BERTimbau e mBERT[13] e seus impactos na performance de tarefas de aprendizado de máquina, os quais revelaram uma correlação entre a segmentação de palavras e a performance de tarefas. Eles avaliaram o modelo em três tarefas de PLN: similaridade textual de sentença, reconhecimento de implicação textual e reconhecimento de entidades nomeadas. O BERTimbau melhora o estado da arte nessas tarefas em relação a modelos multilíngues e abordagens monolíngues anteriores, confirmando a eficácia de grandes modelos de linguagem pré-treinados para o português.

Eles aprimoraram os detalhes da metodologia, especialmente relacionados à geração de vocabulário, estágios de pré-treinamento e ajuste fino. Disponibilizaram o modelo BERTimbau para a comunidade em bibliotecas de código aberto para fornecer

[12] https://paperswithcode.com/dataset/brwac
[13] mBERT é a abreviação para *Multilingual* BERT, sendo uma versão do BERT treinado em múltiplos idiomas, desenvolvida pelo Google AI Language.

linhas de base sólidas para pesquisas futuras e dar mais poder à transferência de linguagem em aplicações de PLN em cenários com limitação de dados rotulados ou insuficientes dados para treinamento de um modelo do zero. Os autores substituíram *WordPiece* pelo formato *SentencePiece* e geraram um vocabulário de 30.000 unidades de subpalavras com o algoritmo BPE[14] e 2 milhões de sentenças aleatórias da *Wikipédia* portuguesa. O vocabulário resultante é ao final convertido em formato *WordPiece* para compatibilidade com o código BERT original. Para o pré-treino, foi utilizado o corpus português aberto com 2,68 bilhões de tokens e 3,53 milhões de documentos, o *Brazilian Web as Corpus* (brWaC), o que garantiu alta diversidade de domínios e qualidade de conteúdo. A utilização somente do corpo do documento (ignorando títulos), com remoção de palavras com erro de codificação e *tags* HTML usando a biblioteca ffty[15], garantiu 17,5 GB de texto bruto (Souza; Nogueira; Lotufo, 2020, p. 4).

A plataforma *Hugging Face*[16] compartilha milhares de modelos de *Transformers* que são utilizados para resolver todo tipo de tarefa de PLN. Várias empresas e organizações utilizam os modelos e contribuem de volta para a comunidade compartilhando seus modelos. Algumas tarefas disponíveis atualmente nos modelos de *Transformers* são elencadas a seguir:

a) Extração de recursos (representação vetorial do texto)
b) Máscara de preenchimento
c) Reconhecimento de Entidades Nomeadas (REN)
d) Perguntas e Respostas (Q&A)
e) Análise de sentimentos
f) Sumarização de textos
g) Geração de textos
h) Tradução
i) Classificação "*zero-shot*"

Desde a introdução do BERT, modelos de linguagem baseados na arquitetura de *Transformers* estabeleceram-se como o padrão para tarefas de PLN (Patil; Gudivada,

[14] BPE significa *Byte Pair Encoding* e se refere a uma técnica eficaz para tokenizar texto que ajuda a lidar com palavras desconhecidas combinando pares de bytes mais frequentes para formar subpalavras.
[15] A biblioteca ftfy é uma ferramenta em Python usada para corrigir e normalizar texto que pode ter problemas de codificação, caracteres Unicode malformados ou outro tipo de corrupção de texto (https://ftfy.readthedocs.io/en/latest/).
[16] https://huggingface.co

2024). Na última década, surgiram diversos modelos que utilizam corpora de pré-treinamento de dimensões cada vez maiores e incrementam significativamente o número de parâmetros. Esta evolução culminou na classificação destes como Modelos de Linguagem de Grande Escala (*Large Language Models*, LLMs). Os LLMs têm demonstrado desempenho "estado da arte" em uma ampla variedade de tarefas, tanto em Compreensão de Linguagem Natural (NLU) quanto em Geração de Linguagem Natural (NLG). Modelos como o *GPT-4* da *OpenAI* e *LlaMA* da *Meta* mostram capacidades de conversação impressionantes. Esses sistemas podem gerar respostas a perguntas feitas em linguagem natural de forma tão coerente e muitas vezes surpreendentes pela inovação que alguns autores até se perguntam se o sistema adquiriu conhecimento (Yildirim; Paul, 2024).

Quadro 6 - Comparativo de alguns dos principais LLM do mercado.

Empresa	Modelo	Janela de Contexto (tokens)	Palavras (aproximadamente)	Páginas de texto (simple prompt)	Parâmetros
Antropic	Claude 2.1	200.000	150.000	470	200 bilhões
OpenAI	GPT-4 Turbo	128.000	96.000	300	1,76 trilhão
Antropic	Claude 2	100.000	75.000	235	130 bilhões
OpenAI	GPT-4 (32K)	32.768	24.576	77	1,76 trilhão
Google	Gemini Pro	32.000	24.000	75	1,8 and 3,25 bilhões (Nano)
Aberto (Google)	PaLM 2	32.000	24.000	75	540 bilhões
OpenAI	GPT-3.5-turbo	16.385	12.288	39	175 bilhões
OpenAI	GPT-4 (8K)	8.192	6.144	20	1,76 trilhão
Aberto (Meta)	Llama 2	4.096	3.072	10	7, 13 e 70 bilhões
Maritaca AI	MariTalk 1.0 small	4.096	3.072	10	175 bilhões
Aberto (TII)	Falcon	2.048	1.536	5	7, 40 e 180 bilhões

Fonte: (Souza, 2023)

No Quadro 6, é possível ver as principais características dos principais LLMs do mercado, incluindo a quantidade de parâmetros. A janela de contexto é a número máximo de tokens que o modelo pode levar em conta em uma única operação de processamento ou geração de texto.

5.3 RECONHECIMENTO DE ENTIDADES NOMEADAS (REN)

Complementar à Mineração de Dados que consiste no processo de descoberta e extração de conhecimento em banco de dados (com dados estruturados), a Mineração Textual foca na análise em grandes volumes de texto não-estruturado ou semiestruturado para extrair informações para diversas atividades, e envolve o uso de técnicas de PLN. Os dados não-estruturados são textos desprovidos de organização predefinida ou formato específico, como o texto de um livro, imagens, áudios, vídeos, e-mails etc. Por outro lado, os dados semiestruturados possuem alguma forma de organização, mas não seguem a estrutura rígida de um banco de dados convencional. Exemplos incluem formulários HTML, documentos XML e JSON.

O Reconhecimento de Entidades Nomeadas - REN (*named entity recognition*) é um problema clássico no campo do Processamento de Linguagem Natural, que se refere à identificação de entidades com significado específico em textos não-estruturados e semiestruturados, abrangendo nomes de pessoas, lugares, instituições (organizações) dentre outros. Com o desenvolvimento de algoritmos de aprendizado de máquina e aumento da capacidade de processamento computacional, as pesquisas em REN em domínio geral têm avançado significativamente (MA *et al.*, 2021, p. 260).

Essa tarefa não pode ser realizada simplesmente se fazendo correspondência com *strings* extraídas de dicionários pré-elaborados. Isso porque uma entidade pode ser dependente do contexto. Vejamos, por exemplo, o termo JFK, que pode se referir ao ex-presidente dos Estados Unidos da América *John Fitzgerald Kennedy*, ao Aeroporto Internacional JFK, ou até, eventualmente, a qualquer outra entidade que tenha essa mesma abreviatura (Jiang, 2012, p. 15). No Quadro 7, apresentamos alguns exemplos de entidades.

Quadro 7 - Exemplos de entidades.

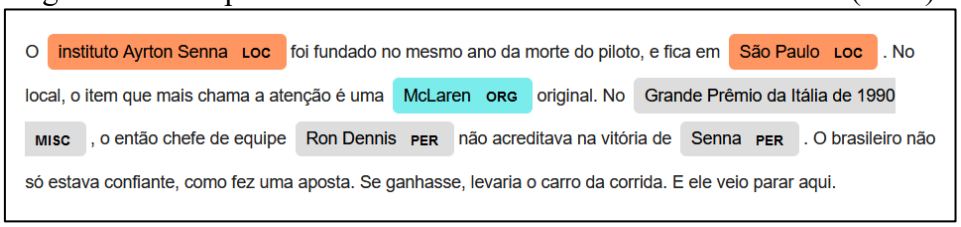
Pessoa	• Arthur Fonseca • Milena Assunção • Thiago Alves
Organização	• Polícia Federal • Partido Novo • UFSC
Local	• Rua Marechal Floriano Peixoto • São José do Rio Preto/SP • Brasil
Tempo	• 03/12/2022 • 1º de janeiro
Valor	• R$ 1.000,00 • Dez reais • US$ 15

Fonte: elaborado pelo autor.

Não obstante a tentativa de designação rígida das entidades, por razões práticas não existe limite para tipos de entidades. A definição dos tipos de entidades a serem utilizados em um projeto depende do objetivo e domínio do trabalho. Um número de telefone pode ser um atributo (característica) de uma entidade Pessoa, ou pode ser uma entidade por si só, a depender da modelagem, do problema a ser pesquisado e ou resolvido, da vontade do arquiteto etc. Entretanto, os tipos mais comuns são pessoa, organização, local, expressões temporais, expressões numéricas, quantidade de unidade e percentuais.

Espera-se que um sistema de extração de entidades ofereça um resultado em que, dada uma sequência de n *tokens* $\{x_1, x_2, \ldots, x_n\}$, o sistema ofereça uma saída tripla (t_s, t_e, k) onde t_s e t_e pertencem a $\{1, \ldots, n\}$, e sejam o *token* de início e *token* de fim, respectivamente, da entidade indicada por k (Souza; Nogueira; Lotufo, 2020, p. 33). Não basta indicar que encontrou a entidade, é preciso indicar sua localização exata no texto. A tarefa do REN é usualmente descrita como uma tarefa de *tag* sequencial, pois para uma sequência de *tokens* $\{x_1, x_2, \ldots, x_n\}$,, o modelo deve apresentar uma sequência de saída de *tags* $\{y_1, y_2, \ldots, y_n\}$,, em que cada *tag* provém de um vocabulário de *tags* pré-definido para um esquema de classe de entidades. Na Figura 7 apresentamos um exemplo de Reconhecimento de Entidades Nomeadas.

Figura 7 - Exemplo de Reconhecimento de Entidades Nomeadas (REN).



Fonte: Trecho extraído da reportagem jornalística e processado pelo autor[17]

Os esquemas de *tagging* mais utilizados são o IOB2 e o IOBES. O primeiro também é denominado BIO em português, que consiste na definição das *tags* B, I e O, seguida da identificação do tipo de entidade, nos dois primeiros casos. A *tag* O é uma marcação que representa a não identificação de entidade para um determinado *token*. A *tag* B indica o início de uma entidade e a *tag* I, que é sempre seguida de outra *tag* B ou I, indica que se trata de continuação da entidade iniciada em B. O esquema IOBES também é conhecido por BILOU e acrescenta algumas *tags* no vocabulário: E-/L- para marcação de final de entidade, e S-/U- para marcação entidade com um *token* simples. O sistema BIO foi originalmente proposto por Ramshaw e Marcus (1995) e também apresentava uma *tag* do nome da entidade: pessoa (PER), organização (ORG), local (LOC) e *miscelaneous* (MISC), que representa entidade que não seja de nenhum tipo anteriormente descrito. Para facilitar a compreensão, apresentamos um exemplo através do Quadro 8, na qual apresentamos *tags* de uma determinada sentença pelos dois esquemas (BIO e BILOU).

O Reconhecimento de Entidades Nomeadas é muito importante para a extração de informações, pois a extração de estruturas mais complexas tais como relações e eventos dependem da assertividade do processo de REN, funcionando como um pré-processamento. O REN também possui muitas outras aplicações, como por exemplo, em pesquisas orientadas a entidades, em que identificar corretamente uma entidade nomeada nos documentos é o primeiro passo para a alta relevância dos resultados de pesquisa.

Quadro 8 - Exemplos de sequenciamento de *tags*.

	Valéria	**Assunção**	**Silva**	**trabalha**	**na**	**Polícia**	**Federal**	**em**	**Brasília**	**/**	**DF**	**.**
BIO	B-PER	I-PER	I-PER	O	O	B-ORG	I-ORG	O	B-LOC	O	B-LOC	O
BILOU	B-PER	I-PER	L-PER	O	O	B-ORG	L-ORG	O	U-LOC	O	U-LOC	O

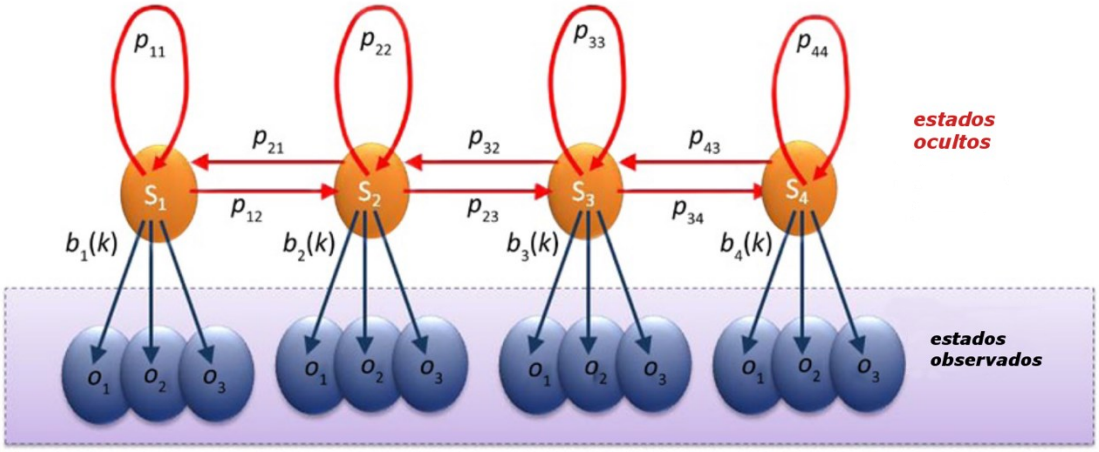
Fonte: elaborado pelo autor.

[17] https://www.band.uol.com.br/noticias/bora-brasil/ultimas/30-anos-da-morte-de-ayrton-senna-o-legado-do-piloto-que-virou-heroi-nacional-16685977

Desde os anos 1990, é possível fazer o reconhecimento de entidades nomeadas com a personalização de regras e modelos feita por especialistas de domínio com precisão considerável, atingindo 90% de assertividade. Esse processo manual, contudo, impede que o reconhecimento seja utilizado em outros domínios, resultando em baixa generalização e portabilidade (Liu; Chen; Xia, 2022, p. 65). No entanto, ao aplicar o aprendizado de máquina (machine learning) no reconhecimento de entidades nomeadas, é possível contornar esses inconvenientes, pois não é preciso construir regras ou modelos manualmente. É possível utilizar um corpus anotado para treinar modelos, incluindo representações como o Modelo Oculto de Markov (HMM - *Hidden Markov Model*) e o Modelo de Campo Aleatório Condicional (CRF - Condicional Random Field).

Um sistema pode ser descrito como estando em um estado específico a um dado momento $\{S_1, S_2, \ldots, S_n\}$. Quando acontece uma transição de um estado S_i para S_j em intervalos de tempo regulares com certa probabilidade p_{ij}, podemos dizer que se trata de um processo estocástico simples (ou seja, depende de uma variável aleatória), no qual a distribuição dos estados futuros depende apenas do estado presente e não dos estados passados. Assim, podemos descrever um processo de Markov discreto como um fenômeno em que um sistema evolui em intervalos regulares, de modo que para um dado estado presente, passado e futuro são estatisticamente independentes. Quando somente o estado presente afeta o estado futuro, podemos dizer que se trata de um sistema dinâmico, o qual nos permite modelar soluções usando variáveis aleatórias, em vez de objetos determinísticos (Awad; Khanna, 2015a, p. 82).

Figura 8 - Modelo oculto de Markov (HMM) com quatro estados ocultos e três estados observados.



p_{ij} = [Modelo de Transição] Probabilidade de transição do estado oculto ***i*** para o estado oculto ***j***
$b_j(k)$ = [Modelo de Observação] Distribuição de probabilidade de observação para o estado ***j***
$k = \{O_1, O_2, O_{3.....} O_n\}$

Fonte: Adaptado de AWAD E KHANNA (2015, p. 84).

No caso de os estados não corresponderem a um fenômeno físico observável, podemos considerar que uma saída observada é uma função probabilística de um estado. Cada estado, então, pode produzir uma série de saídas de acordo com uma distribuição de probabilidade única, e cada saída diferente pode ser potencialmente gerada em qualquer estado. O modelo resultante é o modelo estocástico duplamente embutido, conhecido como HMM, ilustrado na

Figura 8.

O HMM é um modelo estatístico que interpreta um processo não observável (ou oculto), analisando o padrão a partir de uma sequência de atributos observáveis. Esse processo pode então ser inferido indiretamente, prevendo a saída do processo (Awad; Khanna, 2015b).

Em tarefas de processamento de linguagem natural, a sequência de palavras que forma uma sentença é normalmente considerada como o dado observável, enquanto o estado oculto é a representação da informação semântica relacionada à sentença. Os parâmetros do HMM são definidos para maximizar a verossimilhança logarítmica (*log-likelihood*) entre a sentença e a informação semântica. A melhor sequência de estados é a que tem maior verossimilhança com a sentença dada. Os estados são mapeados como *tags* semânticas de forma a criar um classificador de PLN, que pode funcionar muito bem para modelos genéricos (Kanya; Ravi, 2012, p. 3).

O modelo de Campo Aleatório Condicional (CAC) tem sido amplamente utilizado em várias tarefas de PLN, tais como visão computacional, bioinformática e REN. Esse framework de sequência de dados utiliza modelo matemático probabilístico baseado em uma abordagem condicional e tem todas as vantagens do Modelo Exponencial de Markov Máximo (MEMM), com a diferença de ter um modelo exponencial único para uma probabilidade conjunta de uma dada sequência de entrada de rótulos, dada uma sequência de observação. A principal vantagem dos CAC para os outros modelos, como o HMM, segundo Lafferty (2001), é a sua natureza condicional, pois resulta do abrandamento de pressupostos sobre a independência dos estados, necessários para os modelos HMM, para assegurar uma inferência tratável. Portanto, a combinação de técnicas de aprendizado de

máquina, como HMMs e CACs, com métodos tradicionais de REN permite melhorar a precisão e a generalização do reconhecimento de entidades nomeadas em diversos domínios, tornando possível a aplicação desses modelos em uma ampla gama de problemas de PLN.

Não obstante, existe uma propagação de erro durante o treinamento nos modelos com aprendizado de máquina, o que fez com que os estudiosos gradualmente começassem a mudar o foco para o aprendizado profundo (*Deep Learning*) (Liu; Chen; Xia, 2022, p. 66). Em 2011, Collobert propôs uma arquitetura de rede neural multicamada que pode lidar com várias tarefas de PLN com velocidade e precisão, inclusive com reconhecimento de entidades nomeadas (Collobert et al., 2011). O projeto do sistema foi determinado justamente pelo desejo dos autores de evitar ao máximo a engenharia de tarefas específicas. Em vez de explorar os recursos de entrada feitos pelo homem otimizados para cada tarefa, o sistema aprende representações internas com base em grandes quantidades de dados de treinamento não rotulados.

O resultado é que o algoritmo de treinamento descobre representações internas que se mostram úteis para todas as tarefas de interesse. A maior parte da tecnologia de rede neural necessária já havia sido descoberta dez anos antes. No entanto, naquela época não havia infraestrutura computacional suficiente para o treinamento do modelo. Além de resolver o problema da propagação de erro nos modelos de aprendizado de máquina, modelos com *Deep Learning* também diminuem a alta dependência de features. Estas se referem às características ou atributos utilizados para representar os dados de entrada em um modelo de aprendizado de máquina. A diminuição da alta dependência de features significa que os modelos de *Deep Learning* são capazes de aprender representações mais complexas e abstratas dos dados, reduzindo a necessidade de especificar manualmente quais características são relevantes para a tarefa.

Conforme Silva (2021), a Vinculação de Entidade - VE (*entity linking*) é a tarefa de identificar cada entidade única mencionada em um texto não-estruturado e vincular suas menções com as entidades correspondentes em uma base de conhecimento. A primeira parte de uma VE é o reconhecimento de entidade nomeada, que é responsável por identificar a ocorrência de entidades nomeadas em um texto e classificá-las de acordo com categorias pré-definidas. A entidade nomeada é um objeto físico ou abstrato que pode ser identificado por um nome próprio. Esses objetos são classificados como pessoas,

lugares, organizações etc. Números e expressões temporais também podem ser considerados entidades nomeadas. A segunda parte da VE consiste, por sua vez, na Desambiguação das Entidades Nomeadas - DEN (*named entity desambiguation*), que é responsável por ligar corretamente a entidade à base de conhecimento, tanto no caso da palavra ter diferentes significados ou no caso de diferentes palavras terem o mesmo significado.

5.3.1 Extração de Relacionamentos

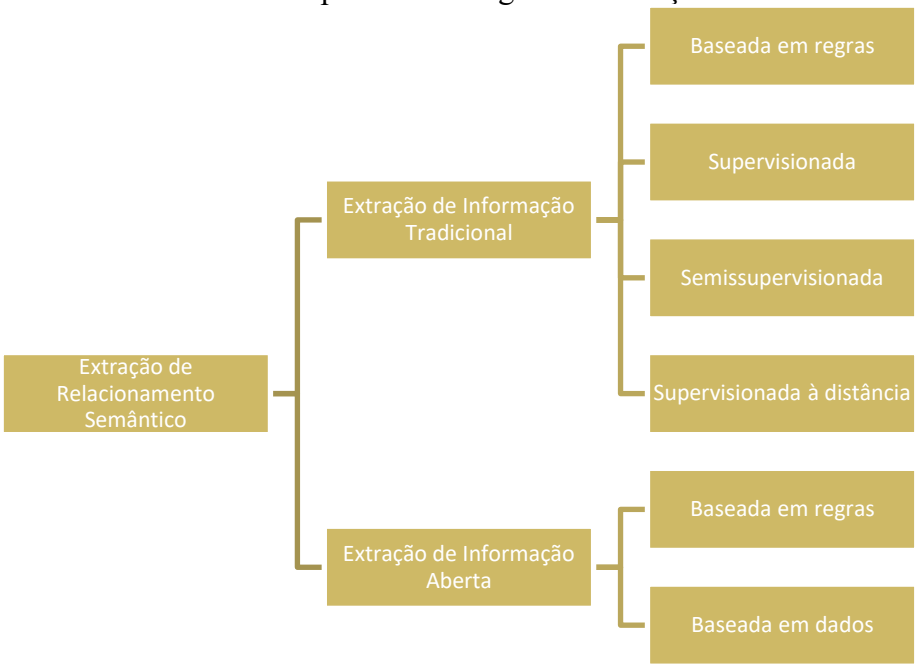
A Extração de Relacionamentos – ER (*relation extraction*) é uma tarefa importante do processamento de linguagem natural que visa identificar e caracterizar relações semânticas entre entidades em um texto, conforme definição de Jiang (2012, p. 22). Ela ainda oferece suporte para outras tarefas como Perguntas e Respostas, Recuperação de Informações, sumarização, anotação semântica da *Web* e construção e extensão de recursos lexicais e ontologias. Por exemplo, identificar a relação entre duas organizações (ORG) através da informação de que "foi adquirida por" ou "foi comprada por". Várias abordagens já foram propostas para ER, incluindo aprendizado de máquina (supervisionado e não-supervisionado), técnicas baseadas em corpora textuais, estratégias linguísticas, recursos como banco de dados lexicais e ontologias, heurísticas baseadas em regras e sistemas híbridos (Abreu; Bonamigo; Vieira, 2013). Na Figura 9, é possível ver a taxonomia proposta por Batista (2016, p. 12) dos vários tipos de abordagens de extração de relacionamentos, as quais podem ser divididas em duas áreas principais, a extração de informação tradicional e a extração de informação aberta.

No ramo de extração de informações tradicionais, inicialmente havia as extrações baseadas em regras, que eram criadas manualmente. Posteriormente, surgiram as abordagens supervisionadas, baseadas em documentos anotados manualmente. Quanto às abordagens semissupervisionadas, elas fazem uso de relacionamentos conhecidos para extrair novos relacionamentos. Por fim, as abordagens supervisionadas à distância fazem uso de base de conhecimento de relacionamentos conhecidos para coletar automaticamente grandes quantidades de dados de treinamento.

O outro ramo de extração de relacionamentos refere-se à extração de informação aberta (*Open Information Extraction* - OIE), que é adequada quando as relações de

destino são desconhecidas e os dados são heterogêneos. A OIE é especialmente utilizada para grandes massas de dados da *Web*, cujas relações de interesse são imprevistas, desconhecidas e em grande número. Podem ser baseadas em dados ou baseadas em regras.

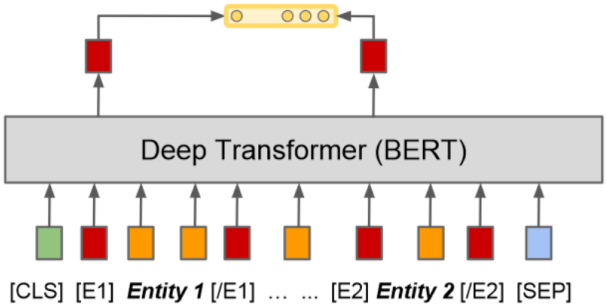
Figura 9 - A taxonomia dos tipos de abordagens de extração de relacionamentos.



Fonte: Adaptada de Batista (2016, p. 12)

Shi e Lin (2019) demonstraram que modelos baseados em BERT podem ser adaptados com simplicidade para extração de relacionamentos (ER) e rotulagem de papel semântico sem recursos (*features*) sintáticos e restrições projetadas por humanos. Em síntese, o modelo de extração de relacionamentos é um classificador que faz a predição da relação r para um dado par de entidades $\{e1, e2\}$. No caso de transformadores, o classificador é adicionado ao topo da saída de estados ocultos. O modelo utilizado é o da arquitetura BERT padrão, com algumas poucas modificações para codificar as declarações de relação de entrada e extrair suas representações de saída pré-treinadas para cálculo de perda e tarefa de ajuste fino. A representação de início de entidade (Figura 10) oferece o melhor desempenho (Soares et al., 2019).

Figura 10 - Marcadores de entidades.



Fonte: Soares et al. (2019)

Métodos que ajustam grandes modelos de linguagem pré-treinados com grandes quantidades de dados rotulados estabeleceram o estado da arte (Yamada et al., 2020). Entretanto, devido a diferentes linguagens e domínios, além do alto custo da anotação humana, existe uma pequena quantidade de conjunto de dados rotulados para aplicação.

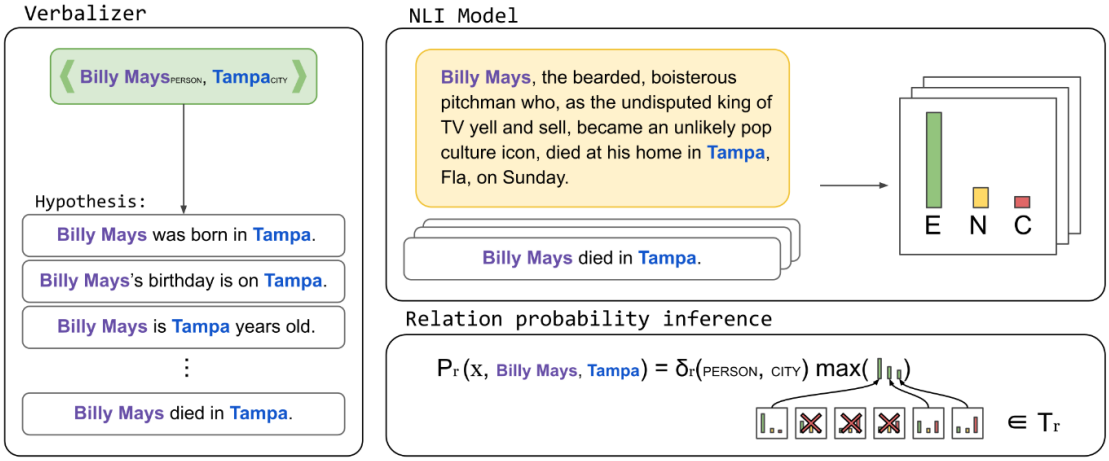
Classificadores de relacionamento baseado na capacidade de transferência de aprendizado como o BERT, que são capazes de extrair relacionamentos sem nenhum dado de treinamento, utilizando técnicas de PLN ou Respostas a Perguntas, surgiram como um novo paradigma e foram descritos como Zero-Shot, com uma alusão ao fato de não precisarem de ajuste fino. Essa abordagem funciona como uma tarefa de vinculação, com verbalizações simples feita à mão. O sistema depende de um mecanismo de implicação textual pré-treinado que é executado como está, por isso o nome Zero-Shot, que significa literalmente "zero tiro", ou ajustado com exemplos rotulados ("poucos tiros") (Sainz et al., 2021). Os autores conseguiram um F1-score de 63% no TACRED[18], que é 17% pontos a mais que o melhor sistema supervisionado nas mesmas condições, e quatro (4) pontos menos que o estado-da-arte, que é pré-treinado com 20 vezes mais dados.

Sainz et al. (2021) reformularam Extração de Relacionamento como sendo uma tarefa de implicação, isto é, dado um texto de entrada contendo duas entidades como premissa e dada uma descrição verbalizada de uma relação como hipótese, a tarefa é inferir se a premissa implica a hipótese de acordo com o modelo NLI. A Figura 11 ilustra as três principais etapas do sistema desenvolvido pelos autores. Uma primeira etapa é focada na verbalização da relação para gerar o conjunto de hipóteses. Na segunda etapa, os autores rodam o modelo e obtém as probabilidades de implicação para cada hipótese.

[18] TACRED é o conjunto de dados mais usado para RE no idioma inglês.

Por fim, com base nas probabilidades e nos tipos de entidades, é retornado um rótulo da relação que tem maior probabilidade de hipótese, incluindo o rótulo *NO-RELATION*, se for o caso.

Figura 11 - Fluxo de trabalho da abordagem ER baseada tarefa de implicação.



Fonte: Sainz et al. (2021)

Importante ressaltar que, ainda hoje, na maior plataforma de repositório de modelos pré-treinados para aprendizado de máquina, o *Hugging Face*, somente existe um modelo inteiramente treinado e ajustado com o idioma Português: BERTimbau. Atualmente está disponível somente para a tarefa de *Fill-Mask*. Apesar disso, existem modelos multilíngue que funcionam bem para outras tarefas em idioma português.

5.4 PRINCIPAIS TÉCNICAS E MÉTRICAS DE AVALIAÇÃO

De acordo com Géron (2021, p. 73), avaliar um sistema classificador é bem mais complicado do que avaliar um regressor. Existem muitos cálculos e medidas de desempenho disponíveis. Neste tópico, apresentaremos algumas delas. Consideramos que VP é o número de verdadeiros positivos, VN o número de verdadeiros negativos, FP o número de falsos positivos e FN o número de falsos negativos. Para comparar o desempenho de sistemas diferentes sobre um mesmo conjunto de dados, é preciso de métricas tipicamente usadas na Recuperação da Informação: precisão, recuperação, F1-score etc. A acurácia é considerada a métrica mais simples de todas e pode ser entendida como a proporção de amostras que foram classificadas corretamente:

$$\text{acurácia} = \frac{(\text{VP} + \text{VN})}{(\text{VN} + \text{VP} + \text{FP} + \text{FN})}$$

A taxa de precisão é a proporção de resultados corretos extraídos pelo sistema, levando-se em consideração todos os resultados extraídos:

$$\text{precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}}$$

A precisão normalmente é usada juntamente com outra métrica chamada de revocação ou taxa de recuperação, que é a proporção dos resultados que foram extraídos corretamente pelo sistema, considerando todos os resultados corretos possíveis:

$$\text{revocação} = \frac{\text{VP}}{\text{VP} + \text{FN}}$$

Essas métricas isoladas muitas vezes não conseguem representar corretamente o modelo e, por isso, é pertinente combinar a precisão e a revocação em uma única métrica denominada F1-score ou pontuação F1 que é média harmônica da precisão e revocação:

$$\text{F1} = \frac{2}{\frac{1}{\text{precisão}} + \frac{1}{\text{revocação}}} = 2 \text{ x } \frac{\text{precisão x revocação}}{\text{precisão} + \text{revocação}}$$

A *F1-score* favorece os classificadores que têm precisão e revocação semelhantes. Entretanto, isso nem sempre é o que se deseja. Em alguns sistemas a preocupação maior é com a precisão e em outros com a revocação. Na detecção de vídeos seguros para crianças, por exemplo, provavelmente se prefere que o classificador rejeite vídeos bons (baixa revocação) e permita apenas os seguros (alta precisão). Neste caso, nenhum vídeo ruim será exibido. Já um classificador para detectar ladrões em lojas por imagem de vigilância, seria bom que o classificador tenha baixa precisão e alta revocação. Haverá muitos falsos positivos, mas quase todos os ladrões serão pegos. Infelizmente não é possível ter as duas coisas. Aumentar a precisão reduz a revocação e vice-versa. Isso se chama trade-off de precisão/revocação (Géron, 2021, p. 75).

Figura 12 - Matriz de confusão.

		Classe prevista	
		Positivos	Negativos
Classe verdadeira	Positivos	**VP**	FN
	Negativos	FP	**VN**

Fonte: elaborado pelo autor.

A melhor forma de se avaliar o desempenho de um classificador é criando uma matriz de confusão (Figura 12). Para calcular a matriz de confusão, primeiro é necessário ter um conjunto de predições de modo que elas possam ser comparadas com as classificações reais. Cada linha de uma matriz de confusão representa uma classe real enquanto cada coluna representa uma classe prevista. Um classificador perfeito teria somente verdadeiros positivos e verdadeiros negativos, isto é, sua matriz de confusão teria valores diferentes de zero apenas na sua diagonal principal (da esquerda superior para a direita superior).

5.4.1 *Corpus* Dourado

Anotar um *corpus* significa marcar um documento, uma frase, palavra ou *token* com uma categoria predefinida. As anotações dependem da finalidade desejada e podem variar desde informações sintáticas até associações semânticas. Dessa forma, os dados textuais são enriquecidos com informações estruturais relevantes. O processamento de linguagem natural é uma tarefa desafiadora porque o significado dos termos depende do contexto. A anotação costuma ser uma etapa de pré-processamento para PLN, e tornou-se essencial para tarefas que dependem de técnicas de ML.

Em 1999, Wissler et al. (2014) propuseram o conceito de *corpus* dourado (GSC[19]) cujo processo de criação definiu como trabalhoso e demorado, o qual é feito manualmente por especialistas. A utilização de corpora textuais rotulados e confiáveis é extremamente importante no desenvolvimento (treino) de algoritmos de processamento de linguagem natural baseados em aprendizado de máquina que usam anotações, uma vez que o erro no corpus é propagado para o sistema final. A rotulação mais confiável é aquela feita à mão, a partir da qual é possível também avaliar rotulações feitas através de algoritmos de rotulação automáticos, denominados de corpus de padrão prata ou prateado (SSC[20]).

Existe ainda a possibilidade de combinação de um processo automático de criação de corpus prateado e um processo manual de criação de corpus dourado, e é isso que caracteriza o processo semiautomático proposto neste trabalho. Um exemplo desse tipo de corpus é o Penn Treebank, que devido à sua origem envolvendo uma combinação de

[19] GSC do inglês que significa *Gold Standard Corpora*
[20] SSC do inglês que significa *Silver Standard Corpora*

rotulação automática e correção manual, além da sua amplitude e qualidade, levou-o a ser reconhecido como um corpus dourado para marcação sintática (Wissler et al., 2014, p. 2).

Desta forma, após a revisão dos conceitos essenciais na literatura, avançamos para a descrição da metodologia utilizada neste estudo.

6 METODOLOGIA

Neste capítulo apresentamos uma visão detalhada dos métodos e técnicas empregados para a consecução dos objetivos propostos, que foram essenciais para a condução da pesquisa. Serão apresentados os procedimentos utilizados para garantir a validade e a confiabilidade dos resultados obtidos.

6.2 CARACTERIZAÇÃO DA PESQUISA

A pesquisa científica é muito importante para o avanço do conhecimento e, consequentemente, para a solução de problemas práticos através de novas tecnologias. Para tanto, é necessário que haja uma delimitação correta do objeto de estudo, com definição clara e precisa do problema a ser investigado (Lakatos; Marconi, 2003). A pesquisa conduzida neste trabalho pode ser classificada de acordo com diferentes critérios, conforme apresentado no Quadro 9.

Quadro 9 - Caracterização da Pesquisa.

Natureza	Trabalho original
Abordagem	Qualitativa
Objetivos	Pesquisa exploratória
Procedimentos técnicos	Pesquisa bibliográfica Pesquisa experimental

Fonte: elaborado pelo autor.

Quanto à natureza da pesquisa, trata-se de um trabalho original, que busca contribuir com o avanço do conhecimento ao apresentar novas descobertas. Quanto à abordagem, é qualitativa. Quanto aos objetivos, podemos classificá-la como exploratória. A pesquisa qualitativa e exploratória se dedica a explorar uma técnica específica por meio de análises qualitativas e estudo de casos. A pesquisa exploratória na Ciência da Informação é uma etapa crucial no processo de investigação que visa explorar, descobrir e compreender fenômenos ainda pouco conhecidos ou pouco estudados dentro dessa área do conhecimento. Lakatos e Marconi (2003) definem pesquisa exploratória como aquela que busca proporcionar maior familiaridade com um problema, com vistas a torná-lo mais explícito ou a construir hipóteses mais precisas. Ela é frequentemente utilizada quando o tema em questão é pouco explorado ou quando se deseja obter uma visão geral sobre um

assunto complexo. Creswell (2009, p. 35) contribui para esse entendimento, explicando que a pesquisa exploratória é útil para identificar variáveis relevantes e estabelecer prioridades na investigação de um determinado problema, quando o tema é novo ou nunca foi abordado com uma determinada amostra. Ela permite ao pesquisador investigar um fenômeno de maneira flexível e aberta, sem a necessidade de estabelecer hipóteses específicas de antemão.

Além disso, a pesquisa não se limita a um único procedimento técnico, mas adota uma abordagem multifacetada. Isso envolve, em primeiro lugar, a pesquisa bibliográfica, considerada um passo fundamental e preliminar para qualquer trabalho científico. Essa etapa visa fornecer ao pesquisador um embasamento teórico sólido e abrangente, permitindo-lhe compreender o estado atual do conhecimento sobre o tema em estudo e identificar lacunas ou áreas pouco exploradas. Adicionalmente, a pesquisa experimental também é empregada. Esta abordagem permite a introdução de novas técnicas ou intervenções controladas pelo pesquisador, visando observar e compreender melhor os resultados obtidos diante de variáveis ambientais cuidadosamente controladas. Constitui-se como oportunidade valiosa para explorar novas possibilidades, testar hipóteses e aprofundar o entendimento sobre fenômenos específicos dentro do campo da Ciência da Informação.

6.3 PROCEDIMENTOS METODOLÓGICOS

A seguir, apresentamos os procedimentos metodológicos utilizados nesta pesquisa, os quais foram orientados à realização dos objetivos específicos. No Quadro 10, são apresentados os procedimentos organizados por objetivos específicos, bem como o produto de cada um.

Quadro 10 - Procedimentos metodológicos para construção do pipeline.

Objetivo Específico	**Procedimento**	**Produto**
(a) Mapear elementos do Reconhecimento de Entidades Nomeadas (REN) e da Extração de Relacionamentos (ER) que sejam úteis para a pesquisa.	(a.1) Investigar os conceitos-chave REN e ER.	Conjunto de conceitos, técnicas e métricas de PLN
	(a.2) Identificar as principais técnicas de REN e ER.	
	(a.3) Executar um levantamento sobre as métricas de avaliação mais utilizadas no PLN, para serem aplicadas em REN e ER.	
(b) Estruturar os elementos de suporte da proposta.	(b.1) Identificar as principais fontes de informação sobre PEPs na Web.	Definições diversas de projeto
	(b.2) Avaliar possíveis métodos de coleta de dados a serem utilizados.	
	(b.3) Delimitar o escopo da proposta.	
	(b.4) Definir o processo de criação de *corpus* dourado.	
	(b.5) Identificar os relacionamentos mais adequados às entidades do escopo selecionado.	
	(b.6) Propor e validar um conjunto de verbalizações úteis para a proposta.	
(c) Desenvolver o pipeline semiautomático baseado em verbalizações.	(c.1) Testar e avaliar a eventual utilização dos elementos mapeados nos objetivos a) e b).	*Pipeline*
	(c.2) Estabelecer o fluxo de atividades necessárias.	
(d) Aplicar a proposta num cenário-teste de investigação policial	(d.1) Definir um cenário de aplicação.	Avaliação do *pipeline*
	(d.2) Selecionar modelos de PLN.	
	(d.3) Escolher as métricas de avaliação apropriadas.	
	(d.4) Executar a proposta	
	(d.5) Avaliar os resultados obtidos.	

Fonte: elaborado pelo autor.

6.3.1 Mapear elementos do Reconhecimento de Entidades Nomeadas (REN) e da Extração de Relacionamentos (ER) que sejam úteis para a pesquisa

Para alcançar os objetivos deste trabalho é essencial mapear elementos de Reconhecimento de Entidades Nomeadas e Extração de Relacionamentos. Para isso, são propostas três ações específicas: investigar conceitos-chave de REN e ER (a.1); identificar as principais técnicas empregadas em REN e ER (a.2); e realizar um levantamento das métricas de avaliação mais utilizadas em PLN, para serem aplicadas em REN e ER (a.3).

A primeira ação (a.1) consiste em investigar os conceitos-chave no Reconhecimento de Entidades Nomeadas e na Extração de Relacionamentos. Esta investigação permitirá compreender os conceitos fundamentais que sustentam as técnicas existentes, identificando os elementos textuais pertinentes ao processo e os diferentes níveis linguísticos envolvidos.

Na sequência, como segunda ação (a.2), é essencial identificar as principais técnicas de REN e ER, para que seja possível, posteriormente, a escolha de uma técnica eficiente para implementação no trabalho.

A terceira ação é investigar as métricas de avaliação (a.3) mais utilizados e adequadas para técnicas de REN e ER, a fim de entender se o processo de criação do pipeline foi bem-sucedido.

6.3.2 Estruturar os elementos de suporte da proposta

A estruturação dos elementos de suporte consiste em uma série de ações destinadas a apoiar o desenvolvimento da proposta. Inicialmente, é necessário identificar a principal fonte de origem dos dados de PEPs na *Web* (b.1). Além disso, avaliar possíveis métodos de coleta de dados a serem utilizados (b.2). A terceira ação (b.3) consiste em delimitar o escopo da proposta. A quarta ação (b.4) cuida de definir o processo de criação do *corpus* dourado. A quinta (b.5) de identificar os relacionamentos mais adequados às entidades do escopo selecionado. E, por fim, a última ação (b.6) envolveu a proposição e validação de um conjunto de verbalizações úteis para a proposta.

Os documentos de textos relacionados a crimes envolvendo PEPs estão disponíveis em fontes abertas (*Web*) através de diferentes sítios de notícias e podem facilmente serem buscados por meio de ferramentas como Google. Assim, a primeira ação de suporte (b.1) consiste justamente em identificar esses sítios para que possam ser coletados os dados de interesse.

A utilização de dados neste trabalho implica não só na identificação de suas origens mais essencialmente da tarefa de realizar a coleta desses dados. Para isso é necessário avaliar possíveis métodos de coleta de dados, através de aplicação de técnica específica, manual ou automática, bem como definir o conteúdo a ser coletado (título, subtítulo, códigos de programação, propagandas de marketing, corpo do texto etc.) e a forma de armazenamento (arquivos de texto, banco de dados etc.).

Em relação ao escopo do trabalho (b.3), é necessário delimitar previamente alguns pontos: definir a quantidade total de documentos de texto a serem coletados, bem como o percentual que será relacionado ao contexto e o percentual que não deverá ser relacionado ao contexto (placebo). Além disso, é necessário fazer uma análise dos documentos pesquisados, preliminarmente, por meio de leitura e avaliação subjetiva para verificar se existe estrutura sintática necessária (as entidades e relação procuradas). Esta avaliação é preliminar, e, em caso de ocorrer uma avaliação incorreta, o documento pode ser posteriormente retirado do corpus e substituído, sem causar prejuízo ao estudo.

A quarta ação (b.4), relacionada à criação de *corpus* textual, envolveu definições sobre como criar o *corpus* dourado. Uma questão importante refere-se à granularidade do processamento do *corpus* textual para extração de relacionamentos: se seriam considerados documentos completos, apenas alguns parágrafos ou somente sentenças. Cada nível de granularidade produz resultados diferentes. Além disso, ao escolher uma abordagem, é necessário criar mecanismos para fornecer e processar o material de entrada de forma adequada, bem como para receber e interpretar os resultados conforme o esperado. Por exemplo, a biblioteca a2t (*ask to transformer*) fornece resultados baseados em pontuações de probabilidade. Ao processar um texto, o resultado pode indicar uma probabilidade de 95% para a existência de um determinado relacionamento em um pedaço de texto (sentença, parágrafo ou texto completo), expressa como uma pontuação (score) de 0.95. Ao processar um texto por inteiro, teremos somente um resultado. Ao processar vários “pedaços” de um texto, teremos vários resultados diferentes e teremos que definir

também uma metodologia para lidar com esses resultados. Além disso, a biblioteca é capaz de fornecer múltiplos resultados, indicando as probabilidades de várias relações avaliadas. Portanto, uma decisão importante é determinar a quantidade de probabilidades desejadas para uma avaliação posterior, ou seja, se se deseja apenas a melhor probabilidade ou se prefere receber as duas ou três principais possibilidades. Portanto, definimos que o processamento seria por sentenças.

A quinta ação (b.5) envolve a definição de quais relações serão usadas neste trabalho bem como, como serão criadas. As relações são o alvo da extração sendo sua definição ato essencial. Neste trabalho selecionamos alguns crimes relacionados ao contexto da proposta para determinação das relações.

Por fim, a quinta e mais importante ação (b.6) é a proposição e validação de um conjunto de verbalizações que sejam eficientes na ER do domínio proposto.

6.3.3 Desenvolver o pipeline semiautomático para ER baseado em verbalizações

O pipeline semiautomático para ER baseado em verbalizações proposto nesta dissertação pode ser dividido em duas grandes partes. A primeira parte consiste na preparação de verbalizações, com suporte de um *corpus* textual, conforme item b.4 do Quadro 9. Este *corpus* textual deve ser coletado da Web, pré-processado com REN, tokenizado e ter seus tokens classificados com sistema IOB (Inside-Outside-Beginning). Ele serve para basear a criação de verbalizações e, através de processamento de ER com ajustes sucessivos (c.1), obter verbalizações devidamente ajustadas para o cenário desejado. A criação de verbalizações é uma necessidade da técnica escolhida, *Zero-Shot Relation Extraction.*

A segunda parte do pipeline consiste em coletar o texto objeto da aplicação, o qual também será pré-processado com REN e, além disso, processado com ER, utilizando as verbalizações criadas na primeira parte.

6.3.4 Aplicar a proposta num cenário de teste de investigação policial

O último objetivo é a aplicação do *pipeline* em um cenário de investigação policial, a fim de demonstrar sua aplicabilidade. Dessa forma, deve-se escolher um cenário específico e proceder a realização das duas grandes etapas do pipeline.

6.4 PROTOCOLO DA REVISÃO SISTEMÁTICA DA LITERATURA

Para fazer a RSL, seguimos as orientações de Kitcheham (2004), partindo de uma estratégia de busca predefinida, para identificar e relatar o máximo possível de literatura relevante. As estratégias de busca bem definidas implicam em um protocolo de revisão bem-feito, que especifica uma questão de pesquisa precisa. O planejamento de uma revisão sistemática da literatura é dividido pela autora em três fases principais: planejamento, condução e relatório. Nos tópicos seguintes faremos a exposição do planejamento, incluindo a revisão do protocolo e o relatório que será apresentado no último tópico (3.3.7).

6.4.1 Questão de pesquisa

Existe metodologia para a extração de informações da *Web* que inclua o reconhecimento de entidades nomeadas e a extração de relacionamentos, aplicada a Pessoas Expostas Politicamente (PEPs) como prática empresarial do tipo *know your customer* (KYC) ou *risk management* ou, ainda, como prática investigativa (persecução criminal) para prevenção à corrupção e lavagem de dinheiro ou financiamento ao terrorismo?

Os tópicos e as palavras-chaves da RSL são apresentados no Quadro 11.

Quadro 11 - Tópicos e palavras-chaves da RSL.

Tópicos	**Palavras-chaves**
1 – PEP	"*politically exposed person*"; "*money laundering*"; "*terrorist financing*"
2 – Mineração de texto	"*text mining*"; "*data science*"; "*data mining*"; "*textual classification*"
3 – KYC	"*know your customer*"; "*risk managment*"

Fonte: elaborado pelo autor.

***String* genérica:**

("*politically exposed person*" OR "*money laundering*" OR "*terrorist financing*") AND ("*text mining*" OR "*data science*" OR "*data mining*" OR "*textual classification*") AND ("*know your customer*" OR "*risk management*")

Para o processo de revisão foi utilizado o programa *StArt*[21], desenvolvido pela *LaPES*, da UFSCAR. Após a pesquisa nas bases selecionadas, obtivemos um retorno de 101 trabalhos de diferentes bases, conforme apresentado na Figura 13.

Tipo e ano de publicação:

Artigos científicos publicados entre 2019 e 2023 (recorte temporal de 5 anos feitos no final de 2023).

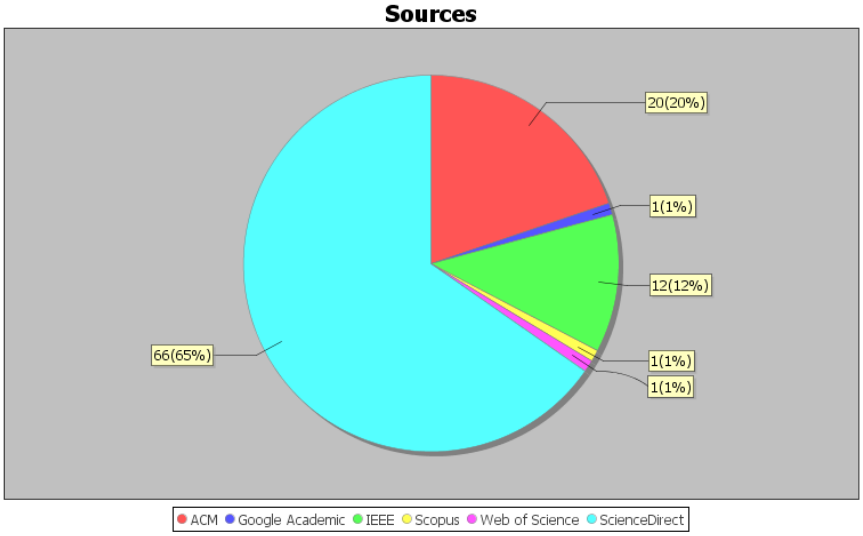
Revisão e seleção de documentos:

Critérios de inclusão (CI):

1. Artigos revisado por pares e publicados entre 2019 e 2023 em inglês ou português;
2. Artigos em que o título e/ou resumo tenha aderência ao tema;
3. Artigos publicados entre os anos de 2019 e 2023;
4. Tipo de documento: Artigos científicos;
5. Artigo que realmente seja aderente ao tema;

[21] https://www.lapes.ufscar.br/resources/tools-1/start-1

Figura 13 - Quantitativo de trabalhos recuperados por base de dados.



Fonte: Extraído do programa StArt.

Critérios de exclusão (CE):

1. Artigos sem aderência ao tema;
2. Artigo com idioma que não seja inglês ou português;
3. Artigos repetidos;
4. Documentos que não sejam artigos científicos;
5. Artigos incompletos ou indisponíveis (sem resumo);
6. Artigos publicados antes de 2019.

6.4.2 Filtragem dos trabalhos recuperados

Foi feita uma análise em diferentes etapas de profundidade para seleção dos trabalhos recuperados nas buscas dos indexadores, aplicando os critérios de inclusão e de exclusão, conforme se vê no Quadro 12. O fluxo quantitativo ao longo das etapas está representado na Tabela 1. Através da Figura 14 é possível ver que somente um único trabalho restou selecionado após a Etapa 3, sendo os demais rejeitados.

Quadro 12 - Etapas do mapeamento bibliográfico.

Etapa	**Critérios de Inclusão**	**Critérios de Exclusão**
Filtragem nas bases de dados	CI1	CE6
Leitura de títulos e resumos	CI2; CI4	CE1; CE2; CE3; CE4; CE5; CE6
Leitura na íntegra	CI5	CE1

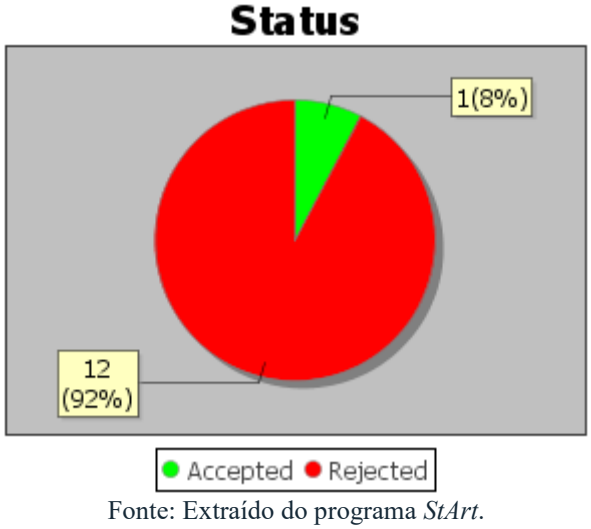
Fonte: elaborado pelo autor.

Tabela 1 - Fluxo quantitativo das etapas de mapeamento bibliográfico.

Base de Dados	Recuperados na Etapa 1	Excluídos na Etapa 2	Incluídos na Etapa 2	Excluídos na Etapa 3	Selecionados na Etapa 3
ACM Digital Library	20	20	-	-	-
ArXiv.org	0	-	-	-	-
Google Scholar	1	-	-	-	-
IEEE Xplore	12	6	6	5	1
Science Direct (via Portal CAPES)	66	59	7	7	0
Scopus	1	1	-	-	-
Springer Link	0	-	-	-	-
Web of Science	1	1	-	-	-

Fonte: elaborado pelo autor.

Figura 14 - Quantitativo de trabalhos aceitos e rejeitados.



Fonte: Extraído do programa *StArt*.

6.4.3 Trabalhos Relacionados

A maior parte dos trabalhos não aderiu à temática da nossa revisão, por não responder à questão de pesquisa e tratar de assunto diverso. Ao longo das etapas, restaram 13 trabalhos para análise final (Etapa 3), dos quais verificamos que 12 deles tratam de monitoramento de transações bancárias, não sendo objeto deste estudo.

Um único artigo foi considerado aceito, pelo qual os autores (THI *et al.*, 2020) desenvolveram um sistema que combina várias técnicas: motor de busca, scraping e pontuação de sentimentos, para apresentar um resultado de avaliação dos clientes no Twitter sobre uma empresa. Basicamente resultado em avaliação se consideram a empresa boa ou ruim. Eles coletaram dados de jornais online e do *Twitter*, via API. Os textos foram pré-processados por meio de tokenização, *tagging*, lematização e transformação de texto bruto em sequências de componentes léxicos. Foi usada a AutoML *Natural Language* do *Google Cloud*. O trabalho propõe uma aplicação para ajudar os banqueiros, seguradoras etc., a identificar seus novos clientes e avaliar o que pode ser oferecido. O serviço proposto é composto de três módulos centrais: módulo de mecanismo de pesquisa, módulo de rastreamento e módulo de análise de sentimento (utilizando mecanismo de IA). A ideia é que o sistema possa potencializar a eficiência da detecção de lavagem de dinheiro com nenhum investimento monetário significativo, minimizando produtivamente seus potenciais reputações no risco e custo.

De fato, sistemas que façam extração de conteúdo da *Web* e análise de forma automatizada (pré-processamento e processamento), especialmente utilizando modelos de PLN, são uma grande oportunidade para alcançarmos um nível de excelência com o aprendizado de máquina. Não só empresas estão ávidas por esses sistemas, mas também órgãos de governo necessitam de tais ferramentas, como é o caso dos órgãos de inteligência de Segurança Pública e de Persecução Criminal.

7 ELEMENTOS DE SUPORTE DA PROPOSTA

Neste capítulo, são expostos diversos elementos que sustentam a proposta, abrangendo delimitações, decisões do projeto e pré-testes requeridos.

7.2 ESCOPO DA PROPOSTA

O presente trabalho tem como objetivo apoiar atividades de inteligência e de investigação policial em trabalhos relacionados a Pessoas Expostas Politicamente, utilizando a análise da extração de relacionamentos através de técnica específica, qual seja, *Natural Language Inference*. Tanto atividades de inteligência quanto investigações policiais envolvem a coleta, análise e produção de conhecimento a partir de fontes abertas, o que representa um desafio devido à quantidade massiva de dados disponíveis e à complexidade no processamento em tempo hábil. Relatórios de inteligência devem ser entregues dentro de prazos estabelecidos, enquanto nas investigações policiais existem limites legais para tramitação e prazos prescricionais que devem ser respeitados.

Para apoiar a criação da proposta, que é baseada em verbalizações contextualizadas, foi criado um *corpus* textual de apoio. Este corpus foi elaborado de forma combinada, utilizando uma abordagem que mistura a automatização e a correção manual, com o objetivo de alcançar maior eficiência.

Além disso, considerando que o foco principal da proposta foi a análise e extração de informações de fontes abertas, é importante ressaltar que, embora a técnica de Extração de Relacionamentos (ER) seja essencial, o desenvolvimento de novos modelos ou técnicas de ER não faz parte deste trabalho. Portanto, foi escolhida uma técnica que não requer treinamento prévio do modelo.

7.3 DADOS PÚBLICOS SOBRE PESSOAS EXPOSTAS POLITICAMENTE

Conforme visto no capítulo de revisão, os dados pessoais de PEPs são consolidados e disponibilizados em fonte aberta mensalmente pela CGU. São dados qualificativos de agentes públicos que desempenham ou tenham desempenhado, nos últimos cinco anos, cargos, empregos ou funções públicas relevantes. Além de um rol de

taxativo de pessoas vinculadas a tais cargos, também são consideradas PEPs seus estreitos colaboradores, definido como qualquer pessoa com estreita relação de conhecimento público com um PEP.

A disponibilização desses dados pela CGU tem como objetivo facilitar o monitoramento de suas atividades, para evitar crimes como lavagem de dinheiro e corrupção. Devido a suas posições de influência, são mais susceptíveis a serem alvo de suborno, corrupção e outras práticas ilícitas.

Neste trabalho, não foi utilizado diretamente os dados de PEPs. A ideia é que a criação do pipeline proposto possa auxiliar a análise de *corpus* textuais extraídos da *Web* para identificar relações suspeitas dessas PEPs.

7.4 ESTRUTURA DO *CORPUS* DOURADO

O *corpus* dourado é um *corpus* textual com anotações feitas de forma manual, o que garante maior confiabilidade dos dados. Entretanto, conforme se vê da literatura, é possível fazer um *corpus* textual de forma mista, combinado um processamento automatizado para a anotação e uma posterior correção manual, garantindo uma melhor eficiência com muito menos tempo de trabalho. Isto porque é muito mais rápido fazer uma revisão manual do que já está anotado ao invés de anotar todo o *corpus*.

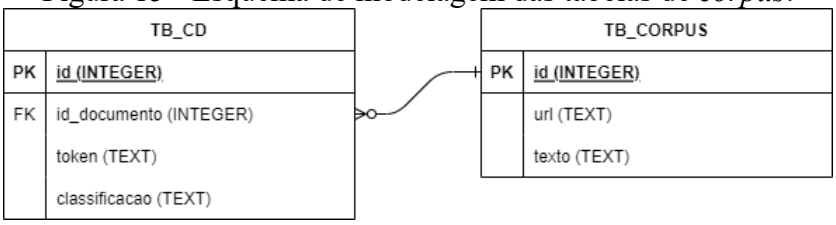
O *corpus* textual utilizado neste trabalho foi constituído de 150 documentos extraídos de sítios da *Web*, os quais foram anotados de forma híbrida, com automatização da anotação e uma posterior revisão manual. A anotação realizada consistiu em Reconhecimento de Entidades Nomeadas, tokenização e classificação de posição (sistema IOB). Por isso a proposta foi classificada de semiautomática, já que combina as duas formas de trabalho. Para a criação do *corpus* textual, a pesquisa dos documentos foi feita em um buscador web específico (Google Notícias), utilizando palavras-chave conexas aos relacionamentos investigados. Não foram utilizados dados pessoais de PEPs.

O cálculo de representatividade do *corpus* textual não foi realizado devido à complexidade em estimar a população total de textos disponíveis na *Web*. A *Web* é um ambiente vasto e dinâmico, onde o volume de conteúdo é constantemente atualizado e expandido, tornando praticamente impossível definir com precisão a "população" de textos sobre qualquer tema específico. Sem uma estimativa clara da população total de

textos, qualquer tentativa de calcular a representatividade do *corpus* coletado pode ser imprecisa e potencialmente enviesada. Além disso, o objetivo da criação do *corpus* textual é apenas obter um pequeno conjunto de dados para fundamentar as verbalizações, evitando baseá-las em suposições ou em ideias do próprio pesquisador. Por tanto, as datas de coleta e publicação dos textos coletados também não foram controladas nem analisadas.

A estrutura de dados sugerida para o *corpus* textual e para o *corpus* dourado foi modelada em banco *SQLite* e apresenta o esquema da Figura 15.

Figura 15 - Esquema de modelagem das tabelas de *corpus*.



Fonte: elaborado pelo autor.

7.5 ENTIDADES E RELAÇÕES A SEREM UTILIZADAS NA PROPOSTA

Para a realização da proposta, escolheu-se uma quantidade limitada de entidades e relações relacionadas ao contexto de crimes envolvendo Pessoas Expostas Politicamente. Como as PEPs podem ser Pessoas Jurídicas, nosso trabalho abrangeu a entidade Organização, além de Pessoa. Entende-se para a entidade Pessoa aquelas relacionadas à Pessoa Física, e para a entidade Organização aquelas relacionadas à Pessoa Jurídica. Além delas, também selecionamos as entidades Local e Tempo para o trabalho, tendo em vista que as relações são ligadas a duas entidades e estas últimas podem favorecer o alcance do resultado.

No Quadro 13, detalhamos as entidades e relações utilizadas neste trabalho.

Quadro 13 - Entidades e relações utilizadas na proposta.

Entidades	**Relações**
Pessoa Organização Local Tempo	Corrupção Operação Lavagem de Dinheiro Sonegação Suborno Tráfico de Influência

Conforme detalhado na Seção 2.5, foi necessário delimitar o escopo do trabalho, e a escolha das palavras-chave foi restrita às seis relações apresentadas no Quadro 13, que abrangem o contexto de tipificações penais relacionadas a crimes envolvendo PEPs.

7.6 MÉTRICAS DE AVALIAÇÃO

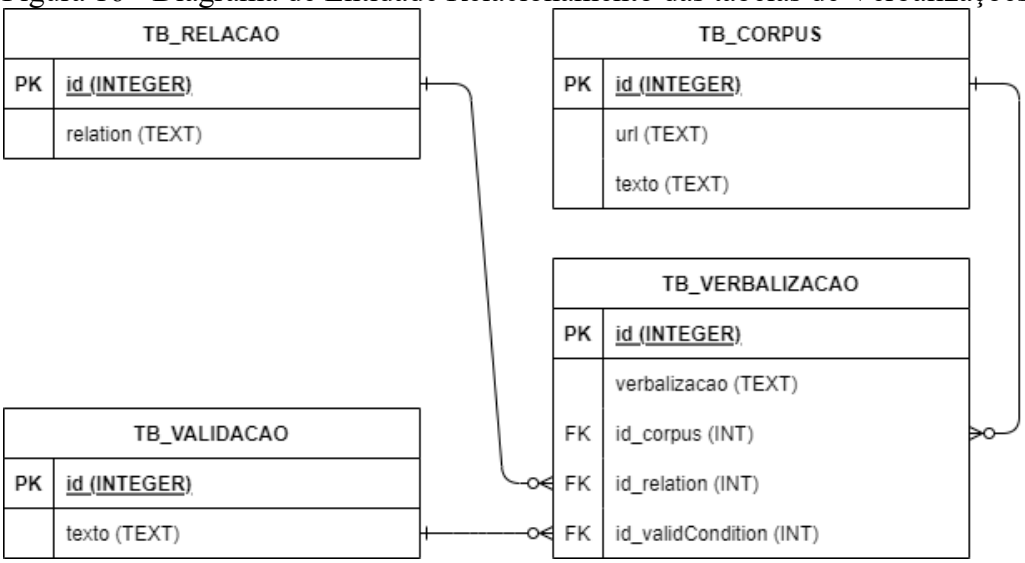
Na etapa de avaliação dos resultados de criação das verbalizações buscou-se métricas usualmente empregadas em modelos de *machine learning* (Klosterman, 2020, p. 82–94). Foi desenvolvido e empregado um código exclusivo (APÊNDICE D), que calcula os valores fundamentais para a construção da matriz de confusão: verdadeiros positivos (VP), falsos positivos (FP), verdadeiros negativos (VN) e falsos negativos (FN). A partir desses valores, foram computadas métricas essenciais, incluindo acurácia, precisão, recuperação e F1-*score*.

7.7 VERBALIZAÇÕES

A criação de verbalizações, conforme descrito por Sainz et al. (2021), é a criação de uma estrutura sintática em que os objetos envolvidos, isto é, as entidades são substituídas por {subj} e {obj}, que representam, respectivamente, o sujeito e o objeto da frase. No entanto, a quantidade textual da verbalização pode ser bem variável. Podemos criar uma verbalização rica, com bastante elemento textual, ou pobre, com pouco elemento textual. Desta forma, a fim de verificar a pertinência de criar verbalizações com muitos elementos textuais e bem específica ou com poucos elementos textuais e mais genérica, realizou-se o pré-teste conforme relato a seguir.

O processo de extração de relacionamentos baseado em verbalizações escolhido emprega a biblioteca a2t (*ask to transformer*), a qual demanda a criação de três conjuntos de dados: relações, condições de validação e as verbalizações propriamente ditas. Por isso, optamos por trabalhar com três tabelas para armazenar esses dados: TB_RELACAO, TB_VALIDACAO e TB_VERBALIZACAO. Um diagrama de Entidade-Relacionamento é apresentado na Figura 16, demonstrando a relação entre essas tabelas.

Figura 16 - Diagrama de Entidade Relacionamento das tabelas de Verbalizações.



Fonte: elaborado pelo autor.

A tabela principal, denominada TB_VERBALIZACAO, armazena a verbalização criada (campo verbalização) e a relaciona com o documento ao qual foi baseado na TB_CORPUS através de uma chave-estrangeira (FK). Além disso, também faz a relação com a TB_RELACAO, indicando qual a descrição da relação, e com a TB_VALIDACAO, para indicar qual a validação aplicável.

7.7.1 Pré-testes

Foi tomado como ponto de partida o texto do Jornal O Globo, de 23/08/2023, o qual denominamos de texto original:

> O criminalista Marcos Vinicius Borges, conhecido como “advogado ostentação”, foi condenado nesta terça-feira a 5 anos e 5 meses de reclusão por tráfico de influência e estelionato praticados contra clientes entre os anos de 2017 e 2018. (O Globo, 2023)

A partir deste texto, criamos dois tipos de relações para o teste, um específico e outro genérico. Ambas as relações são referentes a relacionamento de entidade pessoa com tráfico de influência:

- per:tdinfluencia_especifico
- per:tdinfluencia_generico

Foram criados, consequentemente, dois tipos de verbalizações, cada um associado a uma relação, conforme modelos a seguir:

- per:tdinfluencia_especifico (32 palavras + 2 marcadores de posição):

O criminalista **{subj}**, conhecido como "advogado ostentação", foi condenado **{obj}** a 5 anos e 5 meses de reclusão por tráfico de influência e estelionato praticados contra clientes entre os anos de 2017 e 2018

- per:tdinfluencia_generico (10 palavras + 2 marcadores de posição):

{subj} foi condenado **{obj}** a anos de reclusão por tráfico de influência

As validações de entidades utilizadas foram as mesmas nas duas relações: primeira entidade do tipo PESSOA e a segunda do tipo TEMPO. Portanto, o rótulo da validação resultou em "*PERSON:TIME*". Além do texto original, criamos um texto alternativo, com algumas mudanças nas palavras, mas sem alteração do sentido do texto, conforme acontece nas várias reportagens sobre um mesmo assunto, originárias de fontes diferentes. A ideia é verificar a possível ocorrência de *overfitting*, isto é, uma situação em que o modelo aprende muito bem os dados de treinamento a ponto de replicar seus ruídos e se torna muito bom para os dados para o qual foi treinado, entretanto não é preciso se acontece qualquer variação com os dados, ou *underfitting*, quando o modelo é muito simples para capturar a estrutura dos dados de treinamento e o modelo não consegue aprender os padrões. A seguir, apresentamos o texto alternativo:

> O criminalista Marcos Vinicius Borges, muito conhecido por sua ostentação, foi condenado nesta terça-feira a vários anos de reclusão por tráfico de influência além de outros crimes praticados contra clientes durante 2 anos. (adaptado do Jornal O Globo, 2023)

O processamento de extração de relacionamentos funciona testando hipóteses, de acordo com o modelo pré-treinado de inferência de linguagem natural, em que o texto a ser testado é considerado uma premissa e a verbalização é a hipótese. Desta forma, fizemos uma configuração para que o processamento retorne as três principais hipóteses já que nossos testes envolveram somente duas verbalizações e a hipótese de "*no relation*". Chamamos a hipótese com maior pontuação (melhor probabilidade encontrada) como "*top1 relation*" e a pontuação encontrada como o "top 1 score", e usamos o mesmo padrão

de nomenclatura para as demais hipóteses. O resultado do processamento pode ser visualizado no Quadro 14.

Quadro 14 - Comparação entre cenários de verbalização.

	Top 1		Top 2		Top 3	
	Relation	Score	Relation	Score	Relation	Score
texto original O criminalista Marcos Vinicius Borges, **conhecido como "advogado ostentação"**, foi condenado nesta terça-feira a **5 anos e 5 meses** de reclusão por tráfico de influência **e estelionato** praticados contra clientes **entre os anos de 2017 e 2018**.	Tráfico influência **Genérico**	~0.978	Tráfico influência **Específico**	~0.955	No relation	0
texto alternativo O criminalista Marcos Vinicius Borges, **muito conhecido por sua ostentação**, foi condenado nesta terça-feira a **vários anos** de reclusão por tráfico de influência **além de outros crimes** praticados contra clientes **durante 2 anos**.	Tráfico influência **Genérico**	~0.987	Tráfico influência **Específico**	~0.373	No relation	0

Fonte: elaborado pelo autor.

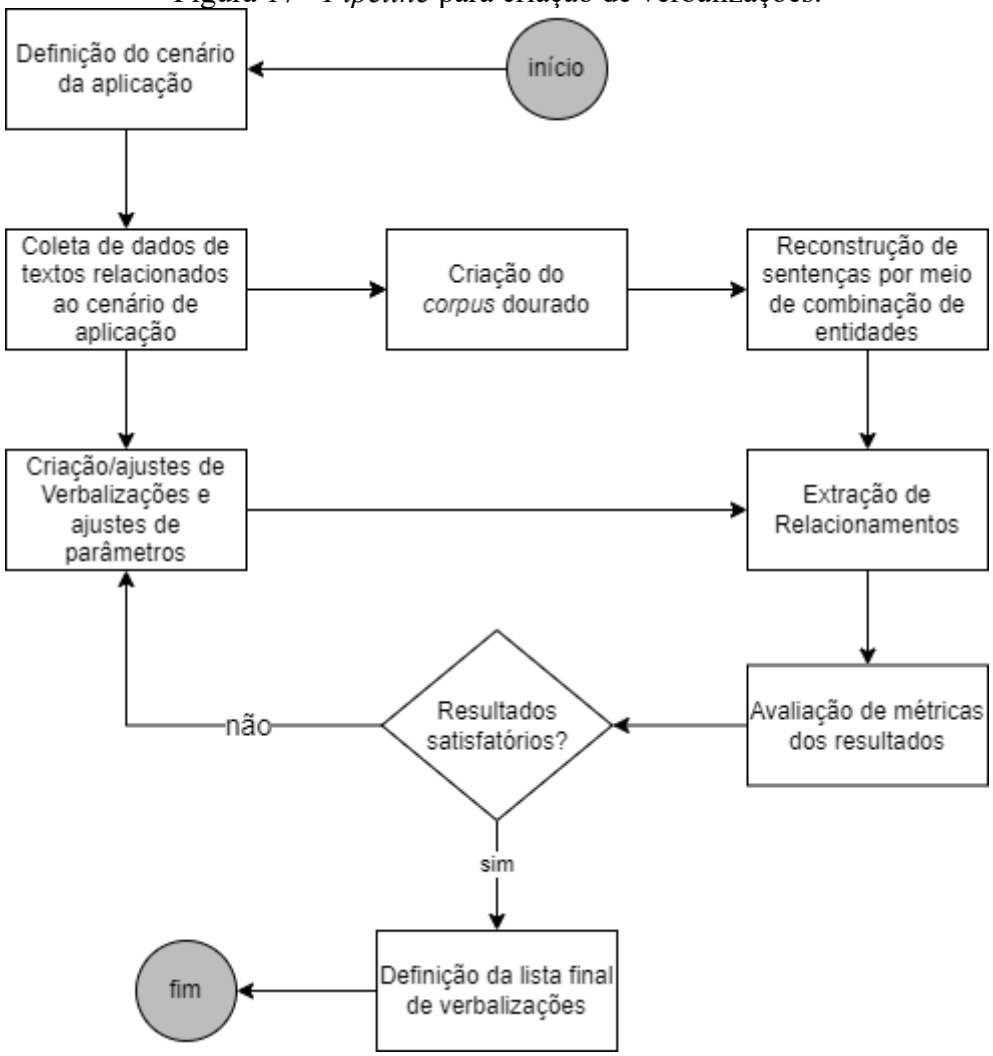
Assim, cada processamento para extração de relacionamentos retorna as três hipóteses mais prováveis (top 1, top 2 e top 3) e suas respectivas pontuações, denominadas scores. Dessa forma, podemos constatar que a versão de verbalização genérica obteve score maior tanto no texto principal quanto no texto alternativo (~0.978 e ~0.987, respectivamente), sendo ligeiramente maior no texto alternativo. Entretanto, a verbalização específica obteve score alto somente com o texto principal (~0.955). O score de verbalização específica com um texto alternativo obteve um valor muito inferior (~0.373), o que indica um provável *overfitting* com a verbalização. A terceira relação encontrada nos dois casos foi "*no_relation*", com *score* 0.0, o que demonstra que o modelo entendeu que havia zero porcento de chance de não ter relação nos textos apresentados.

Desta forma, por meio dos pré-testes foi possível verificar que não é desejável construir a verbalização com muito especificidade, isto é, utilizando o texto base na íntegra, sob pena de acontecer *overfitting*. Constatamos que é necessário escolher as principais palavras da sentença para a construção de uma boa verbalização. No nosso exemplo, a verbalização específica continha 32 palavras, enquanto a verbalização genérica contava somente com 10 palavras e, mesmo assim, obteve um desempenho superior nos testes realizados, tanto com o texto original quanto com o texto alternativo.

8 PROPOSTA DE PIPELINE

A proposta de pipeline consiste na realização de duas grandes etapas, sendo a primeira destinada à criação de verbalizações para ER. Esse processo está representado conceitualmente através da Figura 17, e envolve uma série de ações que estão detalhadas posteriormente na Figura 19.

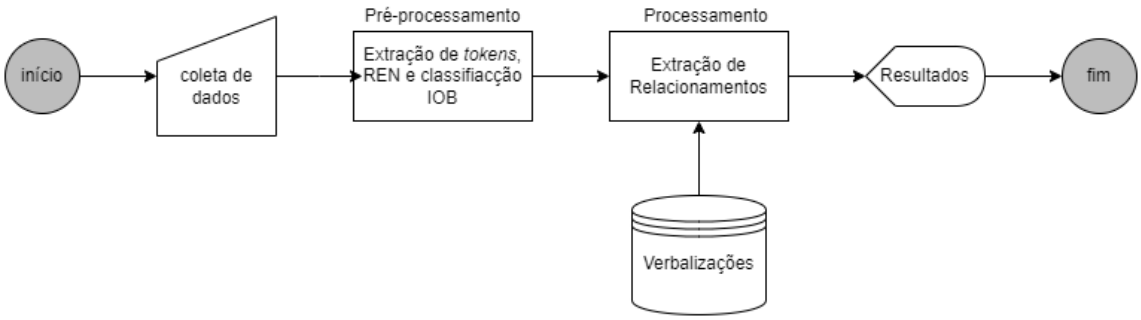
Figura 17 - *Pipeline* para criação de verbalizações.



Fonte: elaborado pelo autor.

Após a criação das verbalizações, é possível então aplicar a ER em qualquer *corpus* textual relacionado ao domínio proposto (segunda etapa), conforme fluxograma apresentado na Figura 18.

Figura 18 - Pipeline para ER baseado em verbalizações.



Fonte: elaborado pelo autor.

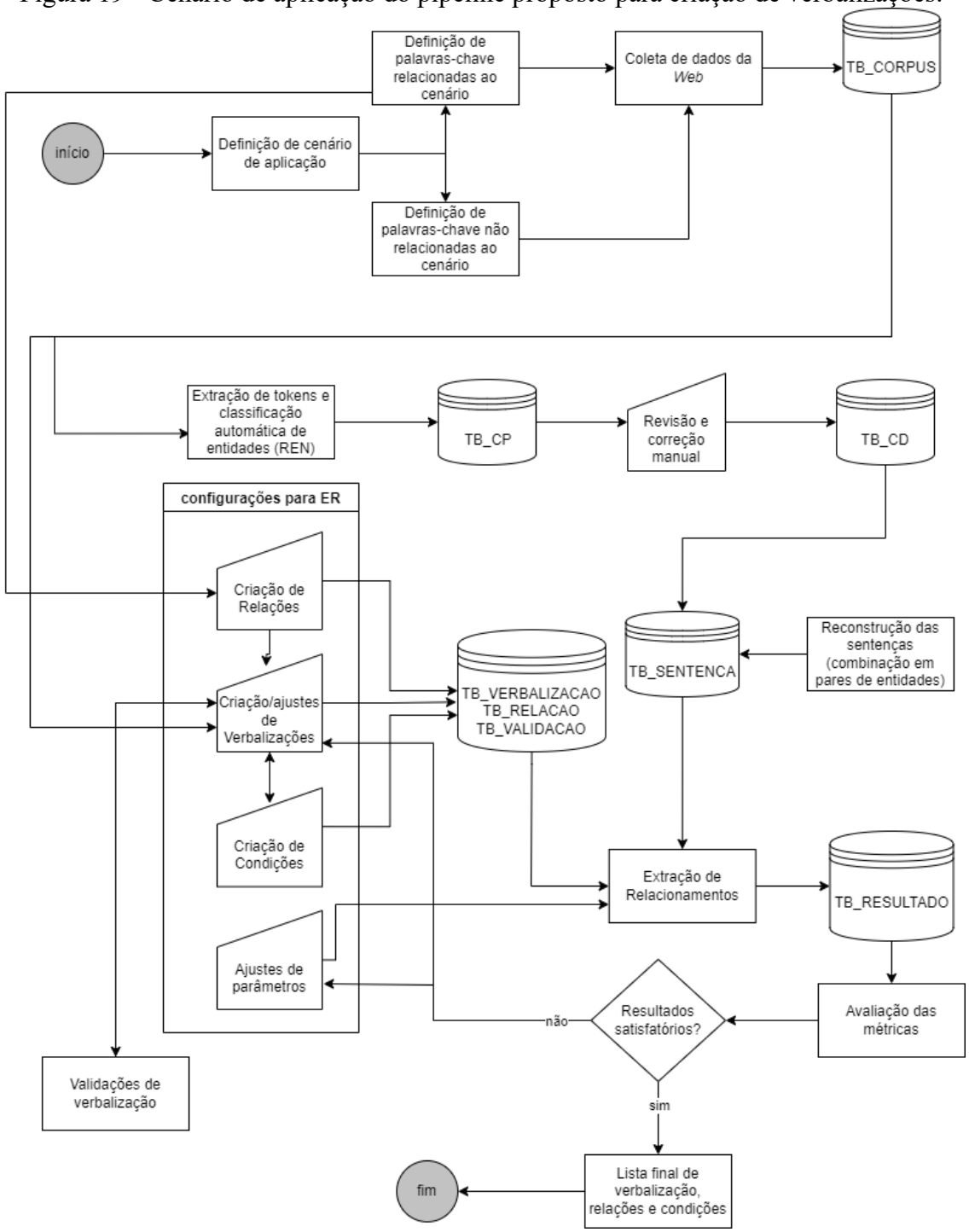
8.2 APLICAÇÃO DA PROPOSTA

De maneira a avaliar o pipeline proposto num cenário real de aplicação, passamos a descrever a aplicação da proposta.

8.2.1 Pipeline

Para o desenvolvimento da primeira etapa da proposta, apresentamos o fluxograma da Figura 19. O processo envolve definições de cenário, coleta de dados, criação de *corpus* textual, criação de *corpus* dourado, criação e ajuste de verbalizações, realização de tarefas de REN e de ER, avaliação de resultados e ajuste de parâmetros, tudo com o propósito final de criar verbalizações específicas para o domínio proposto.

Figura 19 - Cenário de aplicação do pipeline proposto para criação de verbalizações.



Fonte: elaborado pelo autor.

Nas subseções seguintes apresentamos detalhadamente a execução do *pipeline* proposto.

1.1.1 Definição de um cenário de aplicação

De maneira a avaliar o *pipeline* proposto, selecionamos o texto existente na *Wikipédia*, relacionada à Operação Zelotes. Foi escolhido o verbete da *Wikipédia* referente à Operação Zelotes da Polícia Federal. A Operação foi deflagrada em 26 de março de 2015 e teve como objetivo investigar um esquema de corrupção no Conselho de Administração de Recursos Fiscais (CARF), que é um órgão colegiado do Ministério da Fazenda, responsável por julgar recursos administrativos de autuações contra empresas e pessoas físicas por sonegação fiscal e previdenciária[22].

O termo 'zelote' remete a um grupo político judaico que se opunha à dominação romana, especialmente em relação aos impostos cobrados por Roma. No caso da Operação da PF, havia indícios que o esquema manipulava a tramitação de processos e o resultado dos julgamentos, para reduzir ou anular os valores de multas, causando um prejuízo aos cofres públicos que chega a R$ 19 bilhões, considerando todos os processos investigados.

Este cenário nos direciona em como proceder a coleta de dados das etapas seguintes, bem como será o produto final do trabalho.

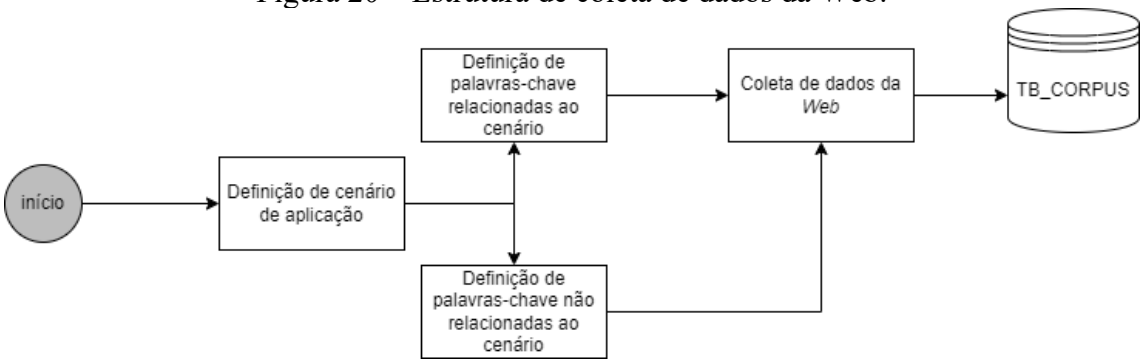
8.2.2 Criação do *corpus* textual

Para criação do *corpus* textual de apoio, iniciamos com a escolha das seguintes palavras-chaves, de acordo com o cenário de aplicação: operação, corrupção, lavagem de dinheiro, sonegação, propina, suborno e tráfico de influência. Na sequência, foram feitas pesquisas de notícias em fontes abertas através da plataforma Google Notícias[23], de forma manual, utilizando as palavras-chaves de modo alternado. Cada sítio com notícia relacionada foi cuidadosamente acessado e lido, a fim de assegurar que se tratava do escopo proposto (análise preliminar). O corpo do documento foi coletado e salvo diretamente no SGBD até completar o total de 100 páginas de notícias. Além disso, também fizemos a coleta de mais 50 documentos sem nenhuma correlação com o domínio, através de pesquisas sobre notícias diversas e aleatórias, utilizando os seguintes

[22] https://pt.wikipedia.org/wiki/Operação_Zelotes
[23] https://news.google.com.br

termos: ciência, tecnologia, artes, lutas, automóveis, saúde, ao qual nós passamos a referir como placebo. Como resultado, obteve-se 150 documentos inseridos em uma tabela denominada TB_CORPUS (cfe. Figura 20).

Figura 20 – Estrutura de coleta de dados da Web.



Fonte: elaborado pelo autor.

Não foram coletados títulos, códigos de programação ou propagandas. Subtítulos dentro do corpo do texto foram coletados. Não houve uma distribuição igualitária de cada assunto, pois a oferta do tema já é naturalmente desequilibrada, com os temas possuindo muito ou pouco conteúdo. Cada texto coletado foi denominado documento e o conjunto dos documentos é o nosso *corpus* textual.

A Figura 21 apresenta os trechos iniciais dos primeiros documentos da tabela TB_CORPUS para ilustração.

Figura 21 - Trechos iniciais dos primeiros 10 documentos da tabela TB_CORPUS que contém 150 documentos.

id	url	texto
Filtro	Filtro	Filtro
1	https://www.cnnbrasil.com.br/politica/...	O depoimento do ex-assessor de Jair Renan Bolsonaro, filho mais ...
2	https://www.cnnbrasil.com.br/economia/...	A construtora Novonor, antiga Odebrecht, ainda deve ao governo d...
3	https://www.cnnbrasil.com.br/politica/...	Líderes da direita no Brasil acreditam que o ex-presidente Jair ...
4	https://www.cnnbrasil.com.br/politica/...	O ministro da Justiça e Segurança Pública, Flávio Dino, usou as red...
5	https://www.cnnbrasil.com.br/nacional/jui...	O delegado Regis Cornelius Celeghini Silveira, titular da 65ª ...
6	https://www.conjur.com.br/2023-out-11/...	O Supremo Tribunal Federal absolveu, por unanimidade, o deputad...
7	https://www.cnnbrasil.com.br/nacional/pf-...	A Polícia Federal (PF) investiga policiais militares do Rio Grande do ...
8	https://www.cnnbrasil.com.br/politica/...	O Supremo Tribunal Federal (STF) condenou, nesta quarta-feira (3...
9	https://www.cnnbrasil.com.br/internacion...	O ex-presidente francês Nicolas Sarkozy perdeu seu recurso contra ...
10	https://www.cnnbrasil.com.br/internacion...	O presidente da Suprema Corte da Ucrânia foi detido por um supost...

Fonte: elaborado pelo autor.

O escopo foi restrito quanto às entidades trabalhadas, focando exclusivamente em Pessoa, Organização, Tempo e Local, tanto na construção do *corpus* quanto na extração de relacionamentos. É crucial destacar que não foi viável acessar uma variedade infinita de conteúdos de diversos sites com temas diversos para este trabalho. A origem específica dos documentos coletados na *Web* não teve relevância para esta pesquisa; o cerne foi a obtenção de textos adequados sobre os temas propostos com as entidades mencionadas.

O código utilizado para REN está disponível no APÊNDICE A, e são necessárias algumas configurações como a indicação do modelo de REN a ser utilizado, indicação de origem dos dados de entrada e destino dos dados de saída (resultado). Os dados de entrada são os documentos da tabela TB_CORPUS e a saída foi armazenada na tabela TB_CP (uma referência a *corpus* prateado).

Utilizou-se o modelo já ajustado do BERTimbau, denominado Luciano/bertimbau-base-lener_br[24] para esse processamento, a partir da base LeNER-Br, que é um dataset de textos legais em português brasileiro que foi formado por 70 textos jurídicos manualmente anotados. As tags usadas foram ORGANIZAÇÃO, PESSOA, TEMPO, LOCAL, LEGISLAÇÃO e JURISPRUDÊNCIA, sendo, portanto, compatível com nosso interesse, que reside somente nas quatro (4) primeiras.

Figura 22 - Resultado parcial do processamento de REN com o sistema BIO (TB_CP).

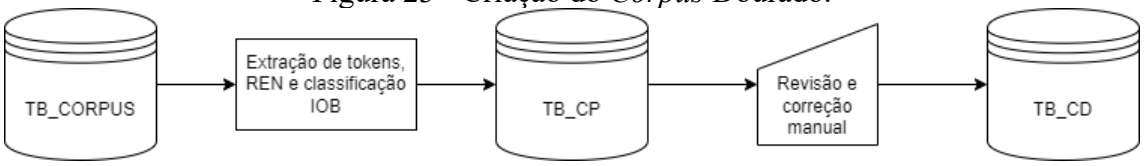
id	id_documento	token	classificacao
Filtro	Filtro	Filtro	Filtro
1	1	[CLS]	O
2	1	O	O
3	1	depoimento	O
4	1	do	O
5	1	ex	O
6	1	-	O
7	1	assessor	O
8	1	de	O
9	1	Jair	B-PESSOA
10	1	Rena	I-PESSOA
11	1	##n	I-PESSOA
12	1	Bols	I-PESSOA
13	1	##ona	I-PESSOA
14	1	##ro	I-PESSOA

Fonte: elaborado pelo autor.

[24] https://huggingface.co/Luciano/bertimbau-base-lener_br

O resultado desse processamento é armazenado na TB_CP (*corpus* prateado) e pode ser visto de forma parcial na Figura 22, consistindo nos campos *id*, *id_documento*, *token* e *classificação*. A classificação utilizada no processo foi feita com o sistema IOB (*Inside-Outside-Beginning*).

Figura 23 - Criação do *Corpus* Dourado.



Fonte: elaborado pelo autor.

A tabela TB_CP é totalmente revisada e corrigida de forma manual para efetivamente se tornar o padrão ouro e oferecer mínimo de possibilidade de erro nas etapas seguintes, para a qual é necessário que sejam entregues as entidades já reconhecidas e classificadas. Após esta última fase de revisão, renomeamos a tabela para TB_CD (*corpus* dourado). O processo completo de criação do *corpus* dourado está ilustrado na Figura 23.

8.2.3 Criação das verbalizações

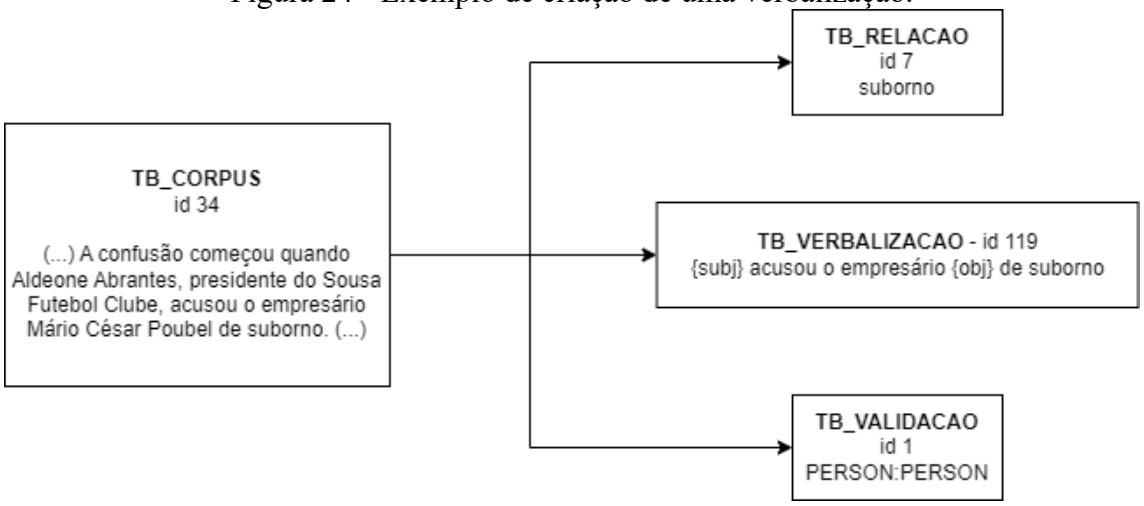
Na tabela TB_RELACAO armazena-se os tipos de relações que foram criadas com base nos termos de pesquisa anteriormente definidos na etapa de coleta de dados: corrupção, operação, lavagem de dinheiro, sonegação, suborno e tráfico de influência. Para as relações de corrupção e operação, decidiu-se dividi-las em dois tipos diferentes cada, para uma melhor especificação, sendo uma relacionada a pessoa e outra relacionada a organização.

Na tabela TB_VERBALIZACAO são salvas as verbalizações propriamente ditas. Todas foram criadas de forma manual, tomando-se como base os documentos coletados na tabela TB_CORPUS, referenciando-os de acordo com cada relação que se deseja fazer processamento posterior de relacionamentos. Cada documento pode ter uma ou mais verbalizações de relações diferentes.

Um exemplo de criação de verbalização pode ser visto na Figura 24. Através dele é possível ver que se utiliza como base um documento coletado e armazenado na

TB_CORPUS com o id nº 34, do qual existe um trecho relacionado a suborno. A partir disso, cria-se a estrutura de verbalização $\{subj\}\ acusou\ o\ empresário\ \{obj\}\ de\ suborno$, salvando-a na TB_VERBALIZACAO com o id nº 199. Além disso, relaciona-se a verbalização com o tipo de relação, que é suborno, por meio da TB_RELACAO, com o id nº 7. Também se cria a respectiva validação, relacionando-a com a tabela TB_VALIDACAO através do id nº 1, o que significa que as duas entidades da verbalização precisam ser do tipo pessoa (person em inglês).

Figura 24 - Exemplo de criação de uma verbalização.



Fonte: elaborado pelo autor.

Assim passa-se a ler cada documento coletado e armazenado na TB_CORPUS, procurando pelo menos uma ocorrência estrutural que contenha as duas entidades e uma relação dentro do escopo. Em alguns documentos não foi possível encontrar uma estrutura adequada, mesmo sendo um texto dentro do domínio/tema. Consequentemente, tivemos que substituir o documento por outro, refazendo as etapas anteriores.

Desta forma, foi realizada a criação manual das verbalizações a partir das estruturas dos documentos do *corpus* textual, bem como foi feito o vínculo com o respectivo documento, conforme modelo de dados (Figura 16). As verbalizações foram inseridas na TB_VERBALIZACAO, com as chaves estrangeiras relacionando-as com a TB_RELACAO e com a TB_VALIDACAO. Como resultado tivemos um total de 246 verbalizações, de oito relações e nove condições de validações. A Figura 25 mostra parte do resultado de verbalização.

Figura 25 - Trecho dos primeiros registros de verbalização (TB_VERBALIZACAO).

id	verbalizacao	id_corpus	id_relation	id_validCondition
Filtro	Filtro	Filtro	Filtro	Filtro
1	segundo apurado pela {obj} investigadores veem {subj} com...	1	1	4
2	{subj} ainda deve ao governo da {obj} por danos causados p...	2	4	2
3	{subj} ainda deve ao governo da {obj} por danos causados p...	2	4	5
4	um escândalo de corrupção como esse seria avassalador par...	3	1	1
5	a {subj} cumpre mandados de busca e apreensão contra o ...	4	2	3
6	{subj} usou as redes sociais nesta {obj} para falar sobre ...	4	5	8
7	{subj} depois de denunciar, por crimes de corrupção o {obj}	5	1	1
8	{subj} absolveu, por unanimidade, o {obj} do crime de ...	6	1	3

Fonte: elaborado pelo autor.

A condições de validações foram criadas de acordo com a necessidade das verbalizações. A validação utilizada consiste nos nomes das entidades em inglês, em caixa alta, separadas por dois pontos, conforme apresentado na Figura 26. A primeira validação feita (id=1) é relacionada à verbalização que possui estrutura compatível com duas entidades do tipo pessoa (“pessoa para pessoa”), razão pela qual foi descrita como sendo “PERSON:PERSON”.

Figura 26 - Tabela TB_VALIDACAO.

id	texto
Filtro	Filtro
1	PERSON:PERSON
2	ORGANIZATION:ORGANIZATION
3	ORGANIZATION:PERSON
4	PERSON:ORGANIZATION
5	ORGANIZATION:LOCATION
6	PERSON:LOCATION
7	ORGANIZATION:TIME
8	PERSON:TIME
9	LOCATION:TIME

Fonte: elaborado pelo autor.

Apresentamos na Tabela 2 o resultado quantitativo das verbalizações criadas, relacionadas ao *corpus* (qtde_corpus) e aos tipos de relações (qtde_verba). Durante o processo de criação das relações, optamos por separar relações de pessoas (per) e de organizações (org), a fim de que possamos verificar o nível de precisão em um eventual processamento de relações específicas.

Tabela 2 - Relações e as respectivas quantidades de *corpus* e verbalizações.

id	relation	qtde_corpus	qtde_verba
Filtro	Filtro	Filtro	Filtro
1	per:corrupcao	27	37
2	per:operacao	15	30
3	org:operacao	13	22
4	org:corrupcao	17	31
5	per:lavagemdinheiro	23	28
6	per:sonegacao	12	14
7	per:suborno	36	59
8	per:traficodeinfluencia	17	25

Fonte: elaborado pelo autor.

A relação do tipo “per:corrupcao” está presente em 27 documentos (itens da TB_CORPUS) dos quais foram baseados a construção de 37 diferentes verbalizações. Assim, ressalta-se que alguns documentos foram origem para diferentes verbalizações, inclusive relacionados a diferentes relações.

8.2.4 Seleção de modelos de PLN

Para aplicação da proposta selecionamos o modelo de ajustado do BERTimbau (Luciano/bertimbau-base-lener_br) para a tarefa de reconhecimento de entidades nomeadas, bem como o modelo Microsoft/DeBERTa-large-mnli, para a Extração de Relacionamentos. Definimos também o limite de negativação de 0,8 e a máquina com GPU RTX4090.

8.2.5 Escolha de métricas de avaliação apropriadas

As métricas de avaliação utilizadas foram as mesmas já detalhadas anteriormente na seção 7.6, quais sejam, acurácia, precisão, F1-score e recuperação, além da matriz de confusão. Além disso, foi necessário fazer uma definição para a leitura e interpretação dos dados, haja vista que foram obtidos vários resultados para cada sentença processada.

Ao consideramos que existem várias pontuações para uma mesma sentença, resultados da ER, decidiu-se tomar como único resultado para efeito de avaliação da aplicação, a melhor resposta de cada sentença que não seja “*no relation*”. Ou seja, foi analisado o processamento que teve o score mais alto de cada sentença e, quando não

houve processamento com score diferente de "*no relation*", foi considerado este último como sendo o resultado.

A seguir, detalhamos o significado e a interpretação de cada uma dessas categorias em relação resultado de processamento do *corpus* textual:

a. VP representam os resultados em que as relações foram corretamente previstas. Isso inclui apenas os documentos com IDs de 1 a 100 da TB_CORPUS, onde as relações foram extraídas. Não há possibilidade de haver VP nos documentos do tipo placebo;
b. FP são classificações para resultados em que houve a classificação prevista incorretamente. Isso abrange as classificações erradas dos documentos com IDs de 1 a 100 da tabela TB_CORPUS, bem como classificações dos demais documentos com IDs de 101 a 150 (placebos), que foram classificados com qualquer relação;
c. VN representam resultados em que não foi prevista nenhuma relação para os documentos do placebo, tendo sido classificados corretamente como "*no relation*" durante o processo de extração de relacionamento; e
d. FN representam aqueles resultados em que o modelo não conseguiu prever nenhuma classificação para os documentos analisados, mesmo que elas existam (somente possível nos documentos de IDs de 1 a 100).

8.2.6 Experimentação de modelos e configurações

Para encontrar o melhor modelo bem como as configurações ideais para utilização das verbalizações, foi necessário fazer processamento de Extração de Relacionamentos e avaliação dos resultados. A etapa de Extração de Relacionamentos utiliza os dados da tabela TB_SENTENCA como entrada e os processa um a um para extração de relacionamentos, através da biblioteca *ask to transformer* (APÊNDICE C). O resultado deve ser salvo na tabela TB_RESULTADO, e as configurações necessárias foram consolidadas na Tabela 3 e são, basicamente, a indicação da entrada de dados, das verbalizações, do modelo de ER e dos documentos de testes. Também dispomos da possibilidade de incluir o conjunto placebo para processamento e ajustar o limiar de negativação, que é o valor do score a partir do qual o modelo considera sucesso na busca

pela relação. Caso o limiar não seja atingido, o primeiro elemento dos resultados de extração de relacionamento será o "*no_relation*".

Tabela 3 - Configurações dos parâmetros utilizados no código proc_a2t.py.

Nome	Tipo	Descrição
mod_er	Texto	indica qual biblioteca de ER será usada
paramlista_verbalizacao	Conjunto de números	indica quais relações será utilizada para o processamento
adiciona_placebo_flag	Lógico	indica se será adicionado *corpus* placebo ao processamento
xproc	Inteiro	número de ref do processamento
n_th	Decimal	limiar de negativação da relação (*negative threshold*)
paramConjuntoMin	Inteiro	indica o primeiro *id* de *corpus*
paramConjuntoMax	Inteiro	indica o último *id* de *corpus*
paramExtracao	Texto	indica qual repositório será usado

Fonte: elaborado pelo autor.

Foram testadas as 1.522 sentenças existentes na tabela TB_SENTENCA, repetidas vezes, com diferentes configurações, conforme demonstrado no Quadro 15. A repetição de processamento se deu em razão da combinação das entidades presentes em cada sentença em pares.

Quadro 15 - Processamentos de ER com parâmetros.

nº proc	mod_er	verbalizacao	nt	corpus	GPU
101	roberta-large-mnli	6	0.7	001-150	RTX4090
102	microsoft/deberta-base-mnli	6	0.7	001-150	RTX4090
103	facebook/bart-large-mnli	6	0.7	001-150	RTX4090
104	roberta-large-mnli	1,2,3,4,5,6,7,8	0.7	001-150	RTX4090
105	microsoft/deberta-large-mnli	6	0.7	001-150	RTX4090
106	microsoft/deberta-large-mnli	1,2,3,4,5,6,7,8	0.7	001-150	RTX4090
107	microsoft/deberta-large-mnli	8	0.7	001-150	RTX4090
108	microsoft/deberta-large-mnli	8	0.8	001-150	RTX4090
109	microsoft/deberta-large-mnli	1,2,3,4,5,6,7,8	0.8	001-150	RTX4090

Fonte: elaborado pelo autor.

Cumpre esclarecer que o código de programa foi criado para apresentar três principais possíveis resultados de relacionamentos (top 1, top 2 e top 3) e suas possíveis probabilidades encontradas, chamadas de scores. No carregamento das relações também é inserido o "*no_relation*" e seu score, que significa que o modelo considera a probabilidade de que o texto não contenha nenhuma relação. Todos os scores possíveis são apresentados em um conjunto de um intervalo fechado entre números reais de 0.0 a 1.0, sendo 0.0 a probabilidade mais baixa (nula) e 1.0 a probabilidade 100%. Para efeito

de melhor controle, criamos um campo denominado “proc” que armazena um número de controle para cada processamento, facilitando processamentos sucessivos e a análise dos resultados.

Ressaltamos que esse processamento com multirrelações, embora não seja um problema de classificação binária clássica, em que o resultado é positivo ou negativo, utilizamos o critério de que, se o modelo previu corretamente pelo menos uma das classes gabaritadas para determinado documento, consideramos que houve sucesso e, portanto, a classificação prevista é verdadeira. Os processamentos para extração de relacionamentos foram realizados sucessivas vezes e numerados conforme o “nº proc”, representado na Tabela 4.

Tabela 4 - Resultado dos processamentos de ER na criação de verbalizações.

nº proc	Modelo de ER	Relação	Tempo (s)	Acurácia	F_1-Score	Recuperação	Precisão
101	roberta-large-mnli	6	949,9	0.713	0.295	0.750	0.184
102	microsoft/deberta-base-mnli	6	563,3	0.380	0.162	0.750	0.091
103	facebook/bart-large-mnli	6	1.080,1	0.133	0.145	0.917	0.079
104	roberta-large-mnli	1,2,3,4,5,6,7,8	51.585,2	0.547	0.688	0.974	0.532
105	microsoft/deberta-large-mnli	6	1.213,3	0.733	0.355	0.917	0.220
106	microsoft/deberta-large-mnli	1,2,3,4,5,6,7,8	Nd*	0.460	0.623	0.985	0.456
107	microsoft/deberta-large-mnli	8	1.938,2	0.753	0.464	0.941	0.308
108	microsoft/deberta-large-mnli	8	1.939,5	0.967	0.865	0.941	0.800
109	microsoft/deberta-large-mnli	1,2,3,4,5,6,7,8	Nd*	0.480	0.632	0.971	0.469

*Nd – não disponível.

Fonte: elaborado pelo autor.

A seguir, apresentamos uma análise detalhada dos resultados, destacando suas características e peculiaridades para uma melhor compreensão.

8.2.6.1 Diferenças entre tamanho dos modelos

Os processamentos de números 102 e 105 foram conduzidos com os mesmos parâmetros, incluindo o uso dos mesmos modelos de NLI. A distinção entre eles residiu no fato de que o processamento nº 102 empregou o modelo básico (base), enquanto o processamento nº 105 utilizou o modelo maior (large). Ao avaliar as métricas obtidas, observou-se que todos os indicadores foram significativamente superiores no processamento com o modelo maior, indicando um desempenho aprimorado.

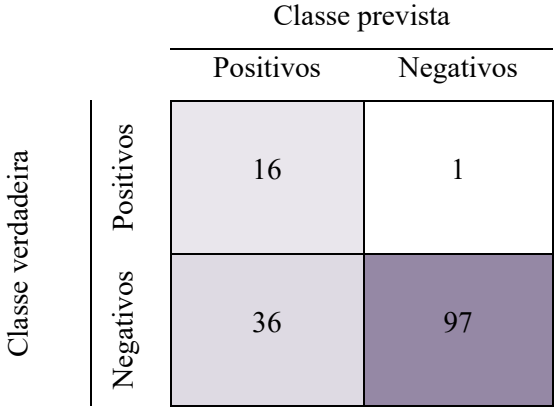
8.2.6.2 Diferenças entre tipos de modelos

Por outro lado, nos processamentos nºs 101, 103 e 105, optamos por empregar três modelos de NLI diferentes, todos na versão *large: roberta-large-mnli, facebook/bart-large-mnli e microsoft/deberta-large-mnli.* Ao analisar os resultados, constatou-se que o modelo da Microsoft, conhecido como DeBERTa, apresentou um desempenho superior em relação aos demais.

8.2.6.3 Desempenho com classificação binária

A seguir, apresentamos os resultados do processamento nº 107, realizado sob o método de classificação binária, no qual foram empregadas exclusivamente as verbalizações da relação tipo 8 (tráfico de influência). A Figura 27 apresenta a matriz de confusão do resultado.

Figura 27 - Matriz de Confusão do processamento da relação 8 (proc nº 107).



Fonte: elaborado pelo autor.

Apresentamos também os indicadores métricos de precisão, recuperação, acurácia e F1-*score* através da Tabela 5.

Tabela 5 - Resultado das métricas do processamento nº 107.

Precisão	Recuperação	F1-Score	Acurácia
~0,308	~0,941	~0,464	~0,753

Fonte: elaborado pelo autor.

Observamos uma boa taxa de recuperação, porém, a precisão foi prejudicada devido ao alto número de Falsos Positivos (FP). No Quadro 16, são apresentadas algumas sentenças que foram classificadas erroneamente como tipo 8, juntamente com seus respectivos scores, para ilustração.

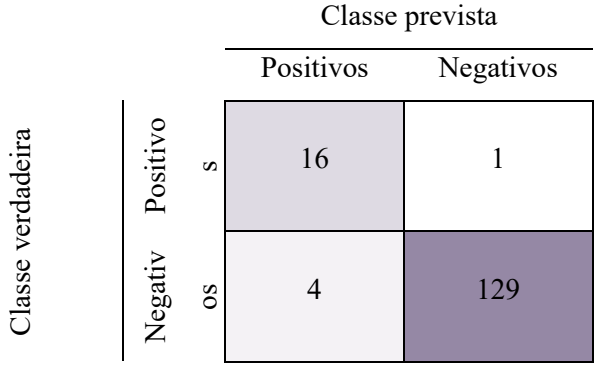
Quadro 16 – Exemplos de sentenças classificadas erradas.

id_corpus	Sentença	score
1	Ele aparece como sócio administrador da **Academia de Tiro 357**, com capital social de R $ 3, 5 milhões, e da **RB Eventos e Mídia LTDA**, com capital social no valor de R $ 105 mil	~0.774
14	Diálogos de Alto Nível O [UNK] painel, Painel Diálogos de Alto Nível, contou com a participação da secretária - executiva da **Controladoria - Geral da União** (CGU), **Vânia Lúcia Ribeiro Vieira**, do diretor da Cepal, Carlos Mussi, e Edson Garutti, coordenador - geral de articulação institucional do Ministério da Justiça (MJSP)	~0.701
16	Flavio Bolsonaro - O pedido de representação foi apresentado pelo ex - deputado **Alexandre Frota**, por supostamente intervir em investigações do Ministério Público do Rio de Janeiro por suposta prática de rachadinha no gabinete do então deputado estadual Flávio Bolsonaro na **Assembleia Legislativa do Rio de Janeiro**	~0.771
112	O **Pacto Internacional sobre Direitos Econômicos, Sociais e Culturais** (PIDESC) é um tratado internacional adotado pela Assembleia Geral das Nações Unidas em 1966 e ratificado pelo Brasil em **1992**.	~0.811

Fonte: elaborado pelo autor.

Ao realizar a consulta no banco de dados para recuperar os registros processados que foram considerados Falsos Positivos (FP), observamos que 177 desses registros foram classificados como relação tipo oito (8) com um score superior a 0,7, que é o nosso limiar para negativação. Desta forma, optou-se em subir o limiar de negativação para 0,8 em um novo processamento registrado como nº 108. Ao construirmos a respectiva matriz de confusão os resultados revelaram que não houve uma grande quantidade de FP (Figura 28).

Figura 28 - Matriz de Confusão do processamento da relação 8 (proc nº 108).



Fonte: elaborado pelo autor.

Após calcularmos as métricas para o processamento nº 108, verificamos que a incidência de FP foi reduzida, o que contribuiu para elevar a acurácia e o F1-Score a níveis melhores que antes (Tabela 6).

Tabela 6 - Resultado das métricas do processamento nº 108.

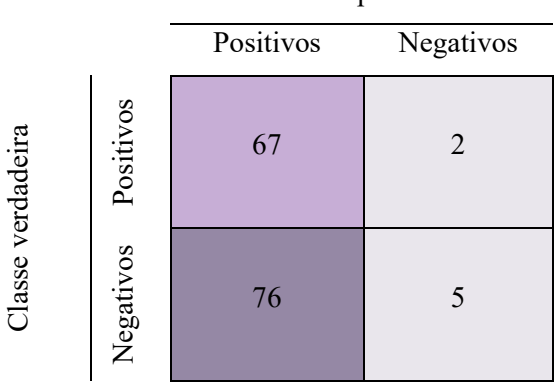
Precisão	Recuperação	F1-Score	Acurácia
~0,800	~0,941	~0,865	~0,967

Fonte: elaborado pelo autor.

1.1.1.1 Desempenho com classificação multicategoria

O processamento nº 109 utilizou todas as classificações (verbalizações) com o *corpus* textual completo, abrangendo documentos com IDs de nº 1 a 150. Utilizou-se o modelo de extração de relacionamento DeBERTa da Microsoft, com um limiar de negativação de 0,8. Os resultados estão disponíveis na Tabela 7 e na Figura 29. Lamentavelmente, ocorreu um reinício não planejado do sistema operacional durante o processamento, impossibilitando o cálculo do tempo de duração. Felizmente, o processo foi retomado sem qualquer outro contratempo.

Figura 29 - Matriz de Confusão do processamento de todas as relações (proc nº 109).



Fonte: elaborado pelo autor.

Tabela 7 - Resultado das métricas do processamento nº 109.

Precisão	Recuperação	F1-Score	Acurácia
~0,468	~0,971	~0,632	~0,480

Fonte: o autor, 2024.

Como observado através dos indicadores e métricas resultantes do processamento que envolveu todas as verbalizações (Figura 29), este não apresentou um desempenho tão satisfatório quanto o obtido no processamento focado em uma única relação (Figura 28).

8.2.7 Execução da proposta em documento de texto

Enfim, tendo sido criadas as verbalizações específicas através do *pipeline* da primeira etapa, passamos a relatar a execução do fluxograma seguinte (segunda etapa), conforme apresentado na Figura 18. O objetivo é a extração de relacionamentos de um documento de texto, verbete da Wikipédia referente à Operação Zelotes da Polícia Federal, conforme definido inicialmente.

Inicialmente foi feita a coleta de dados, copiando o texto diretamente do *Website* da *Wikipedia*, pois se trata de um único documento de conteúdo relativamente pequeno. Colocamos o documento na TB_CORPUS, com o ID nº 151. Em seguida, foi feito o pré-processamento, com a criação do *corpus* prateado, utilizando o código para tokenização, reconhecimento de entidades nomeadas e classificação IOB (APÊNDICE A), tendo sido o resultado salvo na tabela TB_CORPUS_CASO. A partir daí, as sentenças foram reconstruídas e repetidas conforme a quantidade de entidades encontradas, utilizando o

código do APÊNDICE B, o qual foi salvo em uma nova tabela denominada TB_SENTENCA_CASO.

Finalmente, realizamos o processamento de Extração de Relacionamentos, utilizando o código do APÊNDICE C. Para esse processamento, empregamos todas as verbalizações de todas as relações, com um limiar de negativação de 0,8, e com o modelo *microsoft/deberta-large-mnli.*

8.2.8 Avaliação dos resultados

Na Extração de Relacionamentos da primeira etapa da aplicação da proposta foram processadas 47 sentenças diferentes oriundas do *corpus* textual coletado. Cada sentença foi processada várias vezes a depender da quantidade de entidades que continha, uma vez que o pipeline combina as entidades em pares e as oferece para teste de hipóteses no modelo de inferência (extração de relacionamentos).

Em cada sentença, foi feita uma análise manual e foram encontrados 10 casos de FN, isto é, sentenças que foram classificadas como "*no relation*", embora houvesse uma relação existente; 7 casos de VN, aquelas sentenças que foram classificadas como "*no relation*" de forma correta, eis que não havia nenhuma relação existente; 18 casos de VP, aquelas sentenças que foram classificadas corretamente com a relação existente; e 12 casos de FP, aquelas sentenças classificadas com a relação errada. A partir desses dados, calculou-se as métricas apresentadas na Tabela 8.

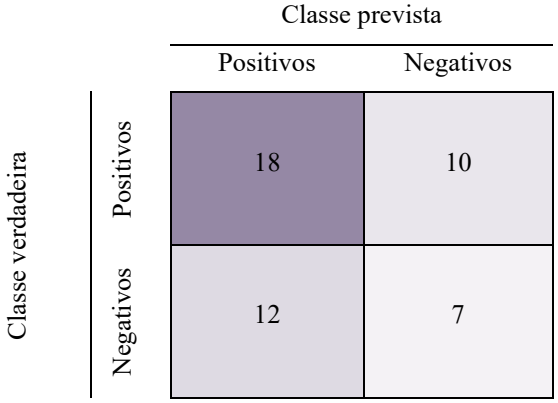
Tabela 8 - Resultado das métricas do processamento nº 111.

Precisão	Recuperação	F1-Score	Acurácia
~0,600	~0,643	~0,621	~0,532

Fonte: elaborado pelo autor.

Além disso, também foi construída a matriz de confusão com os dados do processamento nº 111, apresentada na Figura 30.

Figura 30 - Matriz de Confusão do processamento nº 111.



Fonte: elaborado pelo autor.

A aplicação da proposta pode ser entendida como uma simulação de situação real em que esse pipeline pode ser sistematizado para processar textos oriundos da *Web*. Embora o processamento retorne uma série de pontuações variadas para a mesma sentença, devido à combinação de pares de entidades no pré-processamento, é possível construir regras de negócio, nos mesmos moldes das decisões que tomamos para a avaliação desta proposta, por exemplo, consideração do score mais alto. Também é possível enviar todos os textos de uma só vez para o processamento, caso não haja necessidade de identificação da sentença exata de onde serão encontradas as relações.

8.2.9 Equipamentos e softwares utilizados

Para a construção da proposta, foram utilizados *scripts Python* em ambiente *Windows*. A versão do *Python* utilizada foi a 3.10.11, e a principal biblioteca utilizada a a2t (*ask to transformer*), que já instala dezena de outras bibliotecas por dependência de forma automática, as quais estão descritas no *GitHub*[25]. O Sistema Gerenciador de Banco de Dados (SGBD) utilizado foi o "*DB Browser para SQLite*", versão 3.12.2, com o qual foi possível executar os scripts do *SQLite*, bem como visualizar e inserir alguns dados diretamente no banco de dados.

Este projeto utilizou duas máquinas físicas para os trabalhos (Quadro 17). Conforme era esperado, a máquina com as melhores configurações, especialmente a melhor placa de GPU, obteve desempenho superior. Durante os trabalhos, foi possível

[25] https://github.com/rodrigoraf/ufsc/blob/main/requirements.txt

fazer um teste de um mesmo processamento que foi 14 vezes mais rápido na máquina com GPU superior (Máquina nº 2), a qual se deu prioridade, a partir de então, para realização dos procedimentos.

Quadro 17 - Descrição do equipamento de hardware utilizado.

	Máquina nº 1	**Máquina nº 2**
Processador	11th Gen Intel(R) Core(TM) i5-11400h CPU @ 2.70GHz	Intel(R) Core(TM) i9-10900F CPU @ 2.80GHz
RAM	16,0 GB	64,0 GB
GPU	GEFORCE GTX 1650 4GB	GEFORCE RTX 4090 SG 24GB
SO	Windows 11 Home Single Language versão 22H2	Windows 10 Pro versão 22H2
Tipo de SO	Sistema operacional de 64 bits, processador baseado em x64	Sistema operacional de 64 bits, processador baseado em x64

Fonte: elaborado pelo autor.

Tendo em vista a limitação de infraestrutura das máquinas utilizadas, alguns modelos não puderam ser utilizados, tais como bases extra *large* de alguns modelos. Além disso, alguns processamentos inicialmente previstos para a Máquina 1 precisaram ser realocados para a Máquina 2, para superar a limitação da GPU.

9 CONSIDERAÇÕES FINAS

Este trabalho considerou o crescente papel da identificação de entidades e relacionamentos em investigações policiais e de inteligência, destacando a dispersão de informações valiosas em fontes abertas de PEPs. Assim, delineou-se o objetivo geral de propor um pipeline de mineração textual para extrair relacionamentos de *corpus* textuais visando fornecer insumos para atividade de inteligência e investigação policial. O objetivo foi alcançado, resultando na criação de um cenário favorável para a prospecção de relacionamentos entre entidades, viabilizado pela construção de um *corpus* dourado, a partir da extração manual de elementos textuais da Internet, com base em pesquisas de palavras-chave definidas previamente.

A proposta foi desenhada e implementada em *Python*, através de um pipeline que utiliza a biblioteca a2t, para Inferência de Linguagem Natural. Destaca-se, ainda, que a utilização do modelo DeBERTa da Microsoft, com um limiar de negativação estabelecido em 0.8, apresentou melhores resultados na identificação de relacionamentos em *corpora* textuais relacionados ao domínio proposto: F1-score de 0,865 e acurácia de 0,967, em um cenário de classificação binária. Portanto, os resultados obtidos evidenciam a viabilidade e a eficácia das abordagens e ferramentas empregadas neste estudo, buscando contribuir para a compreensão e análise de relacionamentos entre entidades no contexto de Pessoas Expostas Politicamente (PEPs), especialmente aqueles relacionados à atividade de inteligência e investigação das forças policiais brasileiras. Todos os dados e códigos de programação utilizados, além do *Corpus* Dourado, estão disponíveis no *GitHub*[26].

No entanto, ao longo deste trabalho, nos deparamos com diversas situações que exigem uma análise mais aprofundada. Uma delas refere-se à metodologia de criação do *corpus*, que, embora bem definida e implementada, naturalmente apresenta seus vieses. Apesar de termos conduzido a pesquisa com determinadas palavras-chave e realizado a escolha e a coleta dos documentos dentro do escopo previsto, é importante ressaltar que alguns textos não demonstraram possuir uma estrutura gramatical adequada para as palavras-chave selecionadas. Encontramos textos explicitamente relacionados a um tema específico, mas completamente desprovidos de entidades. Em outros casos, após a coleta,

[26] https://github.com/rodrigoraf/ufsc/

verificou-se que a estrutura gramatical era adequada para outro tema também presente no escopo, levando-o a permanecer no *corpus*. Por exemplo, um texto coletado a partir da pesquisa da palavra-chave “sonegação”, embora não tenha tido uma estrutura gramatical adequada para essa relação, mostrou-se adequado para "corrupção" ou "operação".

Assim, a escolha das relações com semântica muito próxima mostrou-se vantajosa nesse contexto. Embora possa parecer inicialmente desfavorável devido à sobreposição dentro de um mesmo documento, quando consideramos o contexto do escopo – que é a obtenção de relacionamentos envolvendo crimes cometidos por Pessoas Expostas Politicamente –, não encontramos obstáculos significativos. Foi comum encontrarmos o termo “corrupção” nos documentos junto com outras relações, como “tráfico de influência” e “lavagem de dinheiro”, por exemplo. Em geral, optamos por identificar o documento com base no termo mais específico ou, em alguns casos, dividir a verbalização em mais de uma relação. Todas essas escolhas, embora pareçam ocasionalmente improvisadas, foram analisadas e aparentemente não terão impacto significativo capaz de alterar os resultados da técnica apresentada.

A metodologia proposta requer uma cuidadosa elaboração de verbalizações prévias, que constituem o cerne do processo. Uma verbalização inadequada pode resultar em uma extração de relacionamentos de baixa qualidade, tornando essa etapa a mais crítica do trabalho. Embora a técnica não exija treinamento de modelo, as escolhas das verbalizações desempenham um papel crucial. Como observado ao longo do estudo, uma verbalização excessivamente específica pode causar *overfitting*, resultando em desempenho significativamente inferior em outros documentos. Por outro lado, uma generalização excessiva obviamente pode levar ao oposto, gerando muitos falsos positivos (*underfitting*). Portanto, é fundamental criar verbalizações com base em documentos relacionados ao domínio específico de interesse para obter o melhor desempenho possível, o que ressalta a necessidade de criação de *corpus* textual de apoio. Além disso, é necessário realizar testes sucessivos e ajustar as verbalizações.

Por fim, constatamos que este pipeline pode ser adaptado e utilizado para processamento de textos extraídos da *Web* por agentes de segurança e de inteligência, a fim de identificar ações de alvos (investigados) que possam estar passando desapercebidos durante uma investigação. Eventualmente pode ser projetado um sistema que funcione de forma totalmente automatizado, desde a coleta de dados por um

Web Scraping até a apresentação de resultados com trechos em que seja possível extrair relacionamentos de entidades previamente cadastradas.

A atividade de inteligência, assim como as investigações de Polícia Judiciaria, tem um campo vasto no horizonte para coleta e processamento de material oriundo de fontes abertas. O PLN desempenha um papel fundamental nesse processo. A proposição de novas metodologias ou pipelines utilizando as técnicas e ferramentas já disponíveis é um caminho promissor. A partir desses estudos, novas ferramentas podem surgir em auxílio dessas áreas.

Do ponto de vista da Ciência da Informação, este estudo proporcionou uma demonstração da maneira pela qual o processamento da informação pode influenciar a usabilidade do conteúdo. É relevante destacar que o Processamento de Linguagem Natural (PLN) emergiu como uma ferramenta essencial para a utilização de conteúdos obtidos da *Web*. A crescente integração de modelos de Inteligência Artificial (IA) na vida diária da população humana é notável. A criação de novas tecnologias por meio da aplicação da ciência resulta em melhorias significativas na eficiência e no conforto para as pessoas.

No entanto, é importante reconhecer que a limitação na quantidade e diversidade das fontes de dados utilizadas na criação do *corpus* textual pode introduzir um viés na análise dos resultados. Como a seleção das fontes e das palavras-chave foi restrita a um conjunto específico de relações e contextos, há o risco de que certas nuances ou variações de crimes envolvendo PEPs não sejam plenamente capturadas. Esse viés pode impactar a representatividade dos dados e, consequentemente, os resultados.

9.2 TRABALHOS FUTUROS

Este trabalho utilizou um *corpus* textual extraído manualmente da *Web* composto por documentos relacionados a um domínio específico envolvendo PEPs para criar as verbalizações empregadas pela biblioteca principal de extração de relacionamentos. A expansão do *pipeline*, adicionando uma etapa preliminar de coleta de documentos diretamente da *Web* (de forma automatizada) por meio de pesquisas por palavras-chave, poderia resultar em uma implementação interessante.

Considerando que a CGU publica lista com os nomes dos PEPs mensalmente, cada nome poderia ser consultado diretamente em portais de conteúdo, indexadores e mídias sociais. O corpo do documento extraído seria, então, enviado para alimentar o restante do *pipeline*. Além disso, uma classificação de tópicos nos documentos coletados, antes do processamento de extração de relacionamentos, seria relevante para garantir que tópicos irrelevantes não fossem processados.

Outra implementação que poderia ser experimentada para avaliar o impacto no processamento de ER seria a utilização de um modelo de resolução de correferência (CR). Esse modelo tem a capacidade de correlacionar pronomes com substantivos em sentenças adjacentes, substituindo os pronomes pelas entidades correspondentes. Esse processo é realizado por meio de análise de dependências morfológicas, o que poderia resultar em uma melhoria na extração de relacionamentos.

REFERÊNCIAS

AULAR, Y. J. M.; PEREIRA, R. T. Minería de Datos como soporte a la toma de decisiones empresariales. **Opcion, Maracaibo**, v. 23, n. 52, p. 104–118, jan. 2007.

AWAD, M.; KHANNA, R. **Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers**. Berkeley, CA: Apress, 2015a.

AWAD, M.; KHANNA, R. Hidden Markov Model. Em: AWAD, M.; KHANNA, R. (Eds.). **Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers**. Berkeley, CA: Apress, 2015b. p. 81–104.

BAIERLE, I. C.; FROZZA, R.; NARA, E. O. B.; KIPPER, L. M. O ciclo da produção de inteligência como apoio à estratégia de tomada de decisão organizacional. **Revista Produção Online,** [S. l.], v. 11, n. 4, p. 1086–1113, 2011. DOI: 10.14488/1676-1901.v11i4.743. Disponível em: https://www.producaoonline.org.br/rpo/article/view/743. Acesso em: 03 jun. 2023.

BARBARA KITCHENHAM. **Procedures for Performing Systematic Reviews**. UK: Keele University, 2004. Disponível em: https://www.inf.ufsc.br/~aldo.vw/kitchenham.pdf. Acesso em: 31 jan. 2023.

BARBOSA, A. M. A Atividade de Inteligência de Segurança Pública. **Revista Brasileira de Ciências Policiais**, v. 2, n. 1, p. 11. Disponível em: https://dspace.mj.gov.br/bitstream/1/7777/1/RBCP_N1_P11-30.pdf. Acesso em: 20 maio 2012.

BATISTA, D. S. **Large-Scale Semantic Relationship Extraction for Information Discovery**. Doutorado—Lisboa: Universidade de Lisboa, 2016.

BORKO, H. Information science: What is it? **American Documentation**, v. 19, n. 1, p. 3–5, jan. 1968.

BRASIL. **Lei nº 9.613, de 03 de março de 1998**. Dispõe sobre os crimes de "lavagem" ou ocultação de bens, direitos e valores; a prevenção da utilização do sistema financeiro para os ilícitos previstos nesta Lei; cria o Conselho de Controle de Atividades Financeiras - COAF, e dá outras providências. 1998.

BRASIL. **Lei nº 12.850, de 02 de agosto de 2013**. Define organização criminosa e dispõe sobre a investigação criminal, os meios de obtenção da prova, infrações penais correlatas e o procedimento criminal e dá outras providências. Brasília, DF: Diário Oficial da União. 2013.

BRASIL. **Doutrina Nacional de Inteligência de Segurança Pública - DNISP**. 4. ed. rev. e atual. ed. Brasília: Secretaria Nacional de Segurança Pública, 2014.

BRASIL. **Quem somos — Estratégia Nacional de Combate à Corrupção e Lavagem de Dinheiro**. Disponível em: http://enccla.camara.leg.br/quem-somos. Acesso em: 10 jun. 2022a.

BRASIL. **Portal da Transparência - Pessoas Politicamente Expostas**. Disponível em: https://www.gov.br/servidor/pt-br/observatorio-de-pessoal-govbr/portal-da-transparencia-pessoas-politicamente-expostas. Acesso em: 21 fev. 2023b.

BRASIL. **Resolução COAF nº 40, de 22 de novembro de 2021**. Resolução COAF Nº 40, De 22 De Novembro De 2021 - DOU - Imprensa Nacional. . 23 nov. 2021 c, Sec. 1, p. 66.

BUCKLAND, M. Information as thing. **Journal of the American Society for Information Science**, v. v.42, n. n.5, p. p.351-360, 1991.

BUSH, V. As we may think. **Atlantic Monthly**, v. v.176, p. p.101-108, 1945.

CASTELLS, M. **A sociedade em rede.** vol. 1. 8. ed. rev. e ampl. São Paulo: Paz e Terra, 2005.

CEPIK, M. Sistemas nacionais de inteligência: origens, lógica de expansão e configuração atual. **Dados**, v. 46, n. 1, p. 75–127, 2003a.

CEPIK, M. **Espionagem e democracia: agilidade e transparência como dilemas na institucionalização de serviços de inteligência**. 1. ed. Rio de Janeiro: FGV, 2003b.

COAF. **Regulação e Fiscalização**. Disponível em: https://www.gov.br/coaf/pt-br/assuntos/informacoes-as-pessoas-obrigadas/orgaos-reguladores-e-fiscalizadores/regulacao-e-fiscalizacao. Acesso em: 21 fev. 2023.

COAF. **O que faz o COAF?** Disponível em: https://www.gov.br/coaf/pt-br/centrais-de-conteudo/publicacoes/publicacoes-do-coaf-1/o-que-faz-o-coaf-2022-01-24-publicado.pdf. Acesso em: 6 abr. 2024.

COLLOBERT, R. et al. Natural Language Processing (Almost) from Scratch. **NATURAL LANGUAGE PROCESSING**, Journal of Machine Learning Research. v. 12, p. 2493–2537, 2011.

CRESWELL, J. W. **Research design: qualitative, quantitative, and mixed methods approaches**. 3rd ed ed. Thousand Oaks, Calif: Sage Publications, 2009.

DE ABREU, S. C.; BONAMIGO, T. L.; VIEIRA, R. Uma revisão sobre Extração de Relações de olho no Português. **J Braz Comput Soc** 19 , 553–571 (2013). https://doi.org/10.1007/s13173-013-0116-8 Acesso em: 01 nov. 2023.

DEVLIN, J. et al. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. arXiv, , 24 maio 2019. Disponível em: http://arxiv.org/abs/1810.04805. Acesso em: 22 nov. 2022

ENCCLA. **Ações de 2013 — Estratégia Nacional de Combate à Corrupção e Lavagem de Dinheiro**. Disponível em: http://enccla.camara.leg.br/acoes/acoes-2013. Acesso em: 18 fev. 2023.

FERNANDES, F. DO C. Inteligência ou Informações? **Revista Brasileira de Inteligência**, v. v.2, n. n.3, p. p.7-, set. 2006.

GÉRON, A. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn, Keras e TensorFlow:** Conceitos, Ferramentas e Técnicas para a Construção de Sistemas Inteligentes. 2. ed. Rio de Janeiro: Alta Books, 2021.

GONÇALVES, J. B. **A atividade de inteligência e legislação correlata**. Niterói: Impetus, 2010.

HUTCHINS, J. Warren Weaver memorandum July 1949. MT News International. **Anais**... Em: EAMT WORKSHOP: EU AND THE NEW LANGUAGES. Prague, Czech Republic.: European Association for Machine Translation, 1999. Disponível em: https://aclanthology.org/1999.eamt-1.18.pdf. Acesso em: 29 mar. 2024

ISLAM, N. Business Intelligence and Analytics for Operational Efficiency. **SSRN Electronic Journal**, 2018.

JIANG, J. Information Extraction from Text. Em: AGGARWAL, C. C.; ZHAI, C. (Eds.). **Mining Text Data**. Boston, MA: Springer US, 2012. p. 11–41.

KANYA, N.; RAVI, T. Modelings and techniques in named entity recognition: an information extraction task. IET Chennai 3rd International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2012). **Anais**... Em: IET CHENNAI 3RD INTERNATIONAL CONFERENCE ON SUSTAINABLE ENERGY AND INTELLIGENT SYSTEMS (SEISCON 2012). Tiruchengode, India: Institution of Engineering and Technology, 2012. Disponível em: <https://digital-library.theiet.org/content/conferences/10.1049/cp.2012.2199>. Acesso em: 11 nov. 2022

KENT, S. **Informações Estratégicas.** Cel. Hélio Freire. Rio de Janeiro: Biblioteca do Exército, 1967.

KLOSTERMAN, S. **Projetos de Ciência de Dados com Python: Abordagem de Estudo de Caso Para a Criação de Projetos de Ciência de Dados Bem-sucedidos Usando Python, Pandas e Scikit-learn**. São Paulo: Novatec, 2020.

LAFFERTY, J.; MCCALLUM, A.; PEREIRA, F. C. N. **Conditional Random Fields**: Probabilistic Models for Segmenting and Labeling Sequence Data. p. 10, 2001.

LAKATOS, E. M.; MARCONI, M. DE A. **Fundamentos de metodologia científica.** 5 ed. São Paulo: [s.n.].

LIDDY, E. D. Natural Language Processing. Em: **Encyclopedia of Library and Information Science**. 2. ed. [s.l.] NY. Marcel Decker Inc., 2001.

LIU, X.; CHEN, H.; XIA, W. Overview of Named Entity Recognition. **Journal of Contemporary Educational Research**, v. 6, n. 5, p. 65–68, 30 maio 2022.

MA, P. et al. Cybersecurity named entity recognition using bidirectional long short-term memory with conditional random fields. **Tsinghua Science and Technology**, v. 26, n. 3, p. 259–265, jun. 2021.

MINGARDI, G. O trabalho da Inteligência no controle do Crime Organizado. **Estudos Avançados**, v. 21, n. 61, p. 51–69, dez. 2007.

MINISTÉRIO DA JUSTIÇA E SEGURANÇA PÚBLICA. **Enccla 2023:** Estratégia Nacional de Combate à Corrupção e à Lavagem de Dinheiro. Brasília: Secretaria Nacional de Justiça, 2022.

O GLOBO. **'Advogado ostentação' de MT é condenado a 5 anos por tráfico de influência e estelionato**. Disponível em: https://oglobo.globo.com/brasil/noticia/2023/08/23/advogado-ostentacao-e-condenado-a-5-anos-por-trafico-de-influencia-e-estelionato.ghtml. Acesso em: 27 mar. 2024.

PARDO, T. A. S. **PRG0018-101-2023: 3. Níveis de representação e processamento linguístico | e-Disciplinas**. Disponível em: https://edisciplinas.usp.br/mod/resource/view.php?id=4709515. Acesso em: 02 abr. 2024.

PATIL, R.; GUDIVADA, V. A Review of Current Trends, Techniques, and Challenges in Large Language Models (LLMs). **Applied Sciences**, v. 14, n. 5, p. 2074, 01 mar. 2024.

QUADROS, V. **A íntima relação entre narcotráfico e política no Brasil**. **Agência Pública**, 26 out. 2020. Disponível em: https://apublica.org/2020/10/a-intima-relacao-entre-narcotrafico-e-politica-no-brasil/. Acesso em: 18 fev. 2023

RAMSHAW, L.; MARCUS, M. Text Chunking using Transformation-Based Learning. Third Workshop on Very Large Corpora. **Anais**...1995. Disponível em: https://aclanthology.org/W95-0107. Acesso em: 2 abr. 2024

SAINZ, O. et al. **Label Verbalization and Entailment for Effective Zero- and Few-Shot Relation Extraction**. arXiv, , 8 set. 2021. Disponível em: http://arxiv.org/abs/2109.03659. Acesso em: 25 fev. 2023

SARACEVIC, T. **Ciência da informação**: origem, evolução e relações. v. 1, n. 1, p. 22, 1996.

SCHAUFFERT, F. H.; LENTO, L. O. B. **Atividades de Inteligência :** livro didático. 3.ed. Palhoça: UnisulVirtual, 2011.

SHI, P.; LIN, J. **Simple BERT Models for Relation Extraction and Semantic Role Labeling**. arXiv, , 10 abr. 2019. Disponível em: http://arxiv.org/abs/1904.05255. Acesso em: 24 fev. 2023

SILVA, N. E. D. **Extraction of entities and relations in Portuguese from the Second HAREM** Golden Collection. p. 52, 2021.

SOARES, L. B. et al. **Matching the Blanks: Distributional Similarity for Relation Learning**. arXiv, , 7 jun. 2019. Disponível em: <http://arxiv.org/abs/1906.03158>. Acesso em: 26 fev. 2023

SOUZA, A. **Comparando capacidades de LLMs (Large Language Models)**. **Medium**, 30 nov. 2023. Disponível em: https://medium.com/blog-do-zouza/comparando-llms-large-language-models-945c9268c52f. Acesso em: 29 jun. 2024

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. **Portuguese Named Entity Recognition using BERT-CRF**. dez. 2019.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. BERTimbau: Pretrained BERT Models for Brazilian Portuguese. Em: CERRI, R.; PRATI, R. C. (Eds.). **Brazilian Conference on Intelligent Systems.** Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020. v. 12319p. 403–417.

THI, M. H. et al. A Novel Solution For Anti-Money Laundering System. 2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA). **Anais**... Em: 2020 5TH INTERNATIONAL CONFERENCE ON INNOVATIVE TECHNOLOGIES IN INTELLIGENT SYSTEMS AND INDUSTRIAL APPLICATIONS (CITISIA). Da Nang, Vietnam: IEEE, nov. 2020.

VASWANI, A. et al. **Attention is All you Need**. p. 11, 2017. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf. Acesso em: 15 maio 2024.

WISSLER, L. et al. **The Gold Standard in Corpus Annotation**. IEEE GSC 26 jun. 2014. Disponivel em: https://www.academia.edu/23290459/The_Gold_Standard_in_Corpus_Annotation. Acesso em: 15 jun. 2024.

YAMADA, I. et al. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). **Anais**... Em: EMNLP 2020. Online: Association for Computational Linguistics, nov. 2020. Disponível em: <https://aclanthology.org/2020.emnlp-main.523>. Acesso em: 26 fev. 2023

YILDIRIM, I.; PAUL, L. A. From task structures to world models: what do LLMs know? **Trends in Cognitive Sciences**, v. 28, n. 5, p. 404–415, maio 2024.

ANEXO A

DIÁRIO OFICIAL DA UNIÃO

Publicado em: 23/11/2021 | Edição: 219 | Seção: 1 | Página: 66

Órgão: Ministério da Economia/Banco Central do Brasil/Conselho de Controle de Atividades Financeiras

RESOLUÇÃO COAF Nº 40, DE 22 DE NOVEMBRO DE 2021

Dispõe sobre os procedimentos a serem observados, em relação a pessoas expostas politicamente, por aqueles que se sujeitam à supervisão do Conselho de Controle de Atividades Financeiras - Coaf na forma do § 1º do art. 14 da Lei nº 9.613, de 3 de março de 1998.

O PRESIDENTE DO CONSELHO DE CONTROLE DE ATIVIDADES FINANCEIRAS - COAF, no uso da atribuição que lhe confere o inciso IV do art. 9º do Estatuto aprovado pelo Decreto nº 9.663, de 1º de janeiro de 2019, mantido em sua vigência, no que compatível com a Lei nº 13.974, de 7 de janeiro de 2020, na forma do art. 9º da Lei nº 13.901, de 11 de novembro de 2019, torna público que o Plenário do Conselho, em sessão realizada em 7 de outubro de 2021, com fundamento no art. 8º, incisos I, II e IV, do referido Estatuto, e tendo em vista o disposto no § 1º do art. 14 da Lei nº 9.613, de 3 de março de 1998, resolveu:

Art. 1º As pessoas que se sujeitam à supervisão do Conselho de Controle de Atividades Financeiras - Coaf na forma dos arts. 9º e 14, § 1º, da Lei nº 9.613, de 3 de março de 1998, devem adotar as providências previstas nesta Resolução para o acompanhamento de operações ou propostas de operação que envolvam pessoas expostas politicamente.

§ 1º Para fins do disposto nesta Resolução, consideram-se pessoas expostas politicamente:

I - os detentores de mandatos eletivos dos Poderes Executivo e Legislativo da União;

II - os ocupantes de cargo, no Poder Executivo da União, de:

a) Ministro de Estado ou equiparado;

b) Natureza Especial ou equivalente;

c) Presidente, Vice-Presidente e Diretor, ou equivalentes, de entidades da administração pública indireta; e

d) Direção e Assessoramento Superior - DAS de nível 6 ou equivalente;

III - os membros do Conselho Nacional de Justiça, do Supremo Tribunal Federal, dos Tribunais Superiores, dos Tribunais Regionais Federais, dos Tribunais Regionais do Trabalho, dos

Tribunais Regionais Eleitorais, do Conselho Superior da Justiça do Trabalho e do Conselho da Justiça Federal;

IV - os membros do Conselho Nacional do Ministério Público, o Procurador-Geral da República, o Vice-Procurador-Geral da República, o Procurador-Geral do Trabalho, o Procurador-Geral da Justiça Militar, os Subprocuradores-Gerais da República e os Procuradores-Gerais de Justiça dos Estados e do Distrito Federal;

V - os membros do Tribunal de Contas da União, o Procurador-Geral e os Subprocuradores-Gerais do Ministério Público junto ao Tribunal de Contas da União;

VI - os Presidentes e Tesoureiros nacionais, ou equivalentes, de partidos políticos;

VII - os Governadores e Secretários de Estado e do Distrito Federal, os Deputados Estaduais e Distritais, os Presidentes, ou equivalentes, de entidades da administração pública indireta estadual e distrital e os Presidentes de Tribunais de Justiça, Militares, de Contas ou equivalentes de Estado e do Distrito Federal;

VIII - os Prefeitos, os Vereadores, os Secretários Municipais, os Presidentes, ou equivalentes, de entidades da administração pública indireta municipal e os Presidentes de Tribunais de Contas de Municípios ou equivalentes.

§ 2º Para fins do disposto nesta Resolução, também são consideradas pessoas expostas politicamente aquelas que, no exterior, sejam:

I - chefes de estado ou de governo;

II - políticos de escalões superiores;

III - ocupantes de cargos governamentais de escalões superiores;

IV - oficiais generais e membros de escalões superiores do poder judiciário;

V - executivos de escalões superiores de empresas públicas;

VI - dirigentes de partidos políticos.

§ 3º Para fins do disposto nesta Resolução, também são consideradas pessoas expostas politicamente os dirigentes de escalões superiores de entidades de direito internacional público ou privado.

§ 4º Para identificação de pessoas expostas politicamente que se enquadrem no §1º deste artigo ou para confirmação do seu enquadramento em hipótese contemplada em tal dispositivo, devem ser consultadas bases de dados oficiais disponibilizadas pelo Poder Público, a exemplo da relação de pessoas expostas politicamente mantida pela Controladoria-Geral da União - CGU no Portal da Transparência, disponibilizada também pelo Sistema de Controle de Atividades Financeiras - Siscoaf.

§ 5º Para fins de identificação de pessoas expostas politicamente que se enquadrem nos §§ 2º e 3º deste artigo ou para confirmação do seu enquadramento em hipótese contemplada em tais dispositivos, deve-se recorrer a fontes abertas e a bases de dados públicas e privadas.

§ 6º A condição de pessoa exposta politicamente perdura por cinco anos contados da data em que a pessoa deixou de figurar em posição contemplada no § 1º, no § 2º ou no § 3º deste artigo.

Art. 2º As pessoas reguladas pelo Coaf devem dedicar especial atenção às operações ou propostas de operações envolvendo pessoa exposta politicamente, bem como com seus familiares, estreitos colaboradores e ou pessoas jurídicas de que participem, observando, nos casos de maior risco, pelo menos os seguintes procedimentos:

I - obter a autorização prévia do sócio administrador para o estabelecimento de relação de negócios ou para o prosseguimento de relações já existentes;

II - adotar devidas diligências para estabelecer a origem dos recursos;

III - conduzir monitoramento reforçado e contínuo da relação de negócio.

§ 1º Para fins do disposto no caput são considerados familiares os parentes, na linha direta, até o segundo grau, o cônjuge, o companheiro, a companheira, o enteado e a enteada.

§ 2º Para fins do disposto no caput são considerados estreitos colaboradores:

I - pessoas naturais que são conhecidas por terem sociedade ou propriedade conjunta em pessoas jurídicas de direito privado ou em arranjos sem personalidade jurídica, que figurem como mandatárias, ainda que por instrumento particular, ou possuam qualquer outro tipo de estreita relação de conhecimento público com uma pessoa exposta politicamente;

II - pessoas naturais que têm o controle de pessoas jurídicas de direito privado ou em arranjos sem personalidade jurídica, conhecidos por terem sido criados para o benefício de uma pessoa exposta politicamente.

Art. 3º Àqueles mencionados no art. 1º, bem como aos seus administradores, que deixarem de cumprir os deveres disciplinados nesta Resolução serão aplicadas pelo Coaf, cumulativamente ou não, as sanções previstas no art. 12 da Lei nº 9.613, de 1998.

Art. 4º Fica revogada, com a entrada em vigor desta Resolução, a Resolução nº 29, de 7 de dezembro de 2017, do Coaf.

Art. 5º Esta Resolução entra em vigor em 1º de dezembro de 2021.

RICARDO LIÁO

APÊNDICE A

Código de programação: proc_ren.py

```
# Script para sentenciar, tokenizar e classificar documentos
import sqlite3, spacy
from transformers import BertTokenizer, BertForTokenClassification
import torch
modelo = 'Luciano/bertimbau-base-lener_br'
global paramCorpus
# Atenção: Antes de alterar o paramCorpus, certifique que a tabela foi criada.
paramCorpus = 'TB_CORPUS'    # tabela de dados do Corpus. Padrão: TB_CORPUS
banco_origem = sqlite3.connect('corpus.db')
cursor_origem = banco_origem.cursor()
sql = "SELECT id, texto FROM “+paramCorpus
cursor_origem.execute(sql)
result_origem = cursor_origem.fetchall()
# Carregar o modelo BERT pré-treinado
model = BertForTokenClassification.from_pretrained(modelo)
# Carregar o tokenizer correspondente
tokenizer = BertTokenizer.from_pretrained(modelo)
# Carregar o tokenizer do spacy
nlp = spacy.load("pt_core_news_sm")
# Módulo para extrair entidades
def extrai_entidades(arg):
    # Tokenizar o texto de entrada
    tokens = tokenizer.tokenize(arg)
    # Adicionar tokens especiais de início e fim de sequência
    tokens = ['[CLS]'] + tokens + ['[SEP]']
    # Converter os tokens em IDs de token
    input_ids = tokenizer.convert_tokens_to_ids(tokens)
    # Criar um tensor de entrada
    input_tensor = torch.tensor([input_ids])
    # Obter as previsões do modelo
    outputs = model(input_tensor)
    predictions = torch.argmax(outputs.logits, dim=2)
    # Converter as previsões de volta para rótulos de entidades
    predicted_label_ids = predictions[0].tolist()
    predicted_labels = [model.config.id2label[label_id] for label_id in
predicted_label_ids]
    # Mapear os tokens e os rótulos de entidades resultantes
    entities = []
    for token, label in zip(tokens, predicted_labels):
        entities.append({'word': token, 'entity': label})
    # Imprimir os resultados
    return(entities)
# Código para tratar o resultado da REN juntando tokens da mesma entidade
def extract_named_entities(tokens):
```

```
    named_entities = []
    entity_tags = {}
    i = 0
    while i < len(tokens):
        if tokens[i]['entity'].startswith('B-'):
            entity_label = tokens[i]['entity'][2:]
            entity = tokens[i]['word'].replace('##', '')
            i += 1
            while i < len(tokens) and tokens[i]['entity'] == 'I-' + entity_label:
                if '##' in tokens[i]['word']:
                    entity += tokens[i]['word'].replace('##', '')
                else:
                    entity += ' ' + tokens[i]['word']
                i += 1
            named_entities.append(entity)
            entity_tags[entity] = 'B-' + entity_label
        else:
            i += 1
    return [{'word': entity, 'entity': entity_tags[entity]} for entity in named_entities]

def tratar_lista(items):
    global paramCorpus
    new_items = []
    i = 0
    while i < len(items):
        new_item = ''
        current_item = items[i]

        if current_item['entity'].startswith('B') and i < len(items)-1:
            next_item = items[i+1]
            try:
                next_next_item = items[i+2]
                condicao3 = next_next_item['entity'].startswith('I') and
next_next_item['word'].startswith('##') and next_item['entity'][2:] ==
next_next_item['entity'][2:]
            except:
                next_next_item = 0
                condicao3 = False
            condicao1 = next_item['entity'].startswith('I') and current_item['entity'][2:] ==
next_item['entity'][2:]
            condicao2 = next_item['word'].startswith('##') and next_item['entity'][2:] ==
current_item['entity'][2:]
            if condicao1 or condicao2:
                if next_item['word'].startswith('##'):
                    new_word = current_item['word']+next_item['word'].replace('##','')
                else:
                    new_word = current_item['word']+' '+next_item['word']
                if condicao3:
```

```
                new_word = new_word + next_next_item['word'].replace('##','')
                i += 3
            else:
                i += 2
            new_entity = 'B-'+current_item['entity'][2:]
            new_item = {'word': new_word, 'entity': new_entity}
            new_items.append(new_item)
            continue
        new_items.append(current_item)
        i += 1
    return(new_items)

for id, texto in result_origem:
    print(str(id))
    # sentenciar para fazer o REN
    for x in nlp(texto).sents:
        if len(x) <= 2:
            continue
        sentenca = x.text.replace("'"," ").replace('"',"")
        try:
            res = extrai_entidades(sentenca)
        except:
            continue
        for item in res:
            token = item['word']
            classif = item['entity']
            sql_insert = "INSERT INTO "+paramCorpus+" (id_documento, token,
classificacao) VALUES ("+str(id)+",'"+token+"','"+classif+"');"
            if len(sentenca) >0:
                cursor_origem.execute(sql_insert)
                banco_origem.commit()
banco_origem.close()
```

APÊNDICE B

Código de programação: proc_cd_reconstrucao.py

```
# Script para reconstrução de sentenças
import sqlite3, time, sys
from itertools import permutations
from timeit import default_timer as timer
tempo_inicial = time.time()
vMin = 1
vMax = 150
modelo_ren = 'Luciano/bertimbau-base-lener_br'
refer = 'Extracao'
paramExtracao = 'TB_SENTENCA' # 'ExtracaoCD'
paramCorpus = TB_CORPUS'
def combine_entities_permutacao(sentences_with_entities):
    combined_entities = []
    for sentence_data in sentences_with_entities:
        entities = sentence_data['entities']
        # Gerar todas as combinações possíveis de entidades
        entity_permutations = list(permutations(entities, 2))
        for entity_pair in entity_permutations:
            entity1, entity2 = entity_pair
            # Adiciona o par de entidades à lista
            combined_entities.append({
                'sentence': sentence_data['text'],
                'entity1': entity1,
                'entity2': entity2
            })
    return combined_entities

def combine_entities(sentences_with_entities):
    combined_entities = []
    for sentence_data in sentences_with_entities:
        entities = sentence_data['entities']
        # Combinação 2 a 2 das entidades
        for i in range(len(entities)):
            for j in range(i + 1, len(entities)):
                entity1 = entities[i]
                entity2 = entities[j]

                # Adiciona o par de entidades à lista
                combined_entities.append({
                    'sentence': sentence_data['text'],
                    'entity1': entity1,
                    'entity2': entity2
                })
    return combined_entities
```

```
def extract_entities(data):
    sentences = []
    current_sentence = []
    for token in data:
        if token[1] == '[SEP]':
            if current_sentence:
                sentences.append(current_sentence[1:])  # Remove [CLS] from the beginning
            current_sentence = []
        elif token[1].startswith('##'):
            # Junta com o token anterior sem espaços
            current_sentence[-1] = (current_sentence[-1][0] + token[1][2:],
current_sentence[-1][1])
        elif token[1] in [',', '.']:
            # Junta com o token anterior sem espaços
            current_sentence[-1] = (current_sentence[-1][0] + token[1], current_sentence[-
1][1])
        elif token[1].startswith('I-'):
            # Junta com o token anterior com espaços
            current_sentence[-1] = (current_sentence[-1][0] + ' ' + token[1][2:],
current_sentence[-1][1])
        else:
            current_sentence.append((token[1], token[2] if len(token) > 2 else 'O'))
    # Adiciona a última sentença se não terminou com [SEP]
    if current_sentence:
        sentences.append(current_sentence[1:])  # Remove [CLS] from the beginning
    result = []
    for sentence in sentences:
        text = ' '.join([token[0] for token in sentence])
        entities = []
        prev_entity_type = None
        for token in sentence:
            if token[1] == 'O':
                prev_entity_type = None
                continue
            if token[1].startswith('I-'):
                # Agrupa a entidade ao anterior mantendo apenas a classificação anterior
                try:
                    entities[-1] = (entities[-1][0] + ' ' + token[0], entities[-1][1])
                except:
                    pass
            else:
                entities.append((token[0], token[1]))
        result.append({'text': text, 'entities': entities})
    return result

banco_origem = sqlite3.connect('corpus.db')
cursor_origem = banco_origem.cursor()
```

```
vMax += 1 # incrementa o vMax para a utilização no range.
for id_documento in range(vMin, vMax+1):
    sql = 'SELECT id, token, classificacao FROM '+paramCorpus+' WHERE
id_documento = '+str(id_documento)
    cursor_origem.execute(sql)
    resultado = cursor_origem.fetchall()
    sentences_with_entities = extract_entities(resultado)
    combined_entities = combine_entities_permutacao(sentences_with_entities)
    # Exibindo os pares de entidades
    for pair in combined_entities:
        ventity1 = str(pair['entity1'][0]).replace(',','').replace('.','')
        ventity2 = str(pair['entity2'][0]).replace(',','').replace('.','')
        ventity_type1 = pair['entity1'][1]
        ventity_type2 = pair['entity2'][1]
        sql = "INSERT INTO "+paramExtracao+" (sentenca, mod_ren, entity1, entity2,
entity_type1, entity_type2, referencia, id_corpus) VALUES
('"+pair['sentence']+"','"+modelo_ren+"','"+ventity1+"','"+ventity2+"','"+ventity_type1+
"','"+ventity_type2+"','"+refer+"','"+str(id_documento)+")"
        cursor_origem.execute(sql)
        banco_origem.commit()
    print(id_documento)
banco_origem.close()
tempo_final = time.time()
tempo_execucao = tempo_final - tempo_inicial
print('Tempo total: ',str(tempo_execucao))
```

APÊNDICE C

Código de programação: proc_a2t.py

```
# Script para extração de relacionamentos
from a2t.legacy.relation_classification import NLIRelationClassifierWithMappingHead
from a2t.legacy.relation_classification import REInputFeatures
from sklearn.model_selection import train_test_split
import sqlite3, time, sys
import numpy as np
from timeit import default_timer as timer
import os
os.environ["PYTORCH_CUDA_ALLOC_CONF"] = ""
import torch
global cursor, banco, xserieid, n_th, xproc, mod_er, paramlista_verbalizacao,
paramlista_len
xproc = '108'
xserieid = xproc+'.'
n_th = 0.8
xref = 'PROC='+str(xproc)+', NT='+str(n_th)+', '
adiciona_placebo_flag = False
mod_er = "microsoft/deberta-large-mnli"
#paramlista_verbalizacao = '(2,3)' # operacao
#paramlista_verbalizacao = '(8)' # traf de influ
#paramlista_verbalizacao = '(1,4)' # corrupcao
paramlista_verbalizacao= '(1,2,3,4,5,6,7,8)' # todos
#paramlista_verbalizacao = '(5)' # # lavagem de dinheiro
#paramlista_verbalizacao = '(6)' # sonegacao
#paramlista_verbalizacao = '(7)' # suborno
paramlista_len = len([c for c in paramlista_verbalizacao if c.isdigit()])
paramConjuntoMin = 1
paramConjuntoMax = 150
paramExtracao = 'TB_SENTECA'
banco = sqlite3.connect('corpus.db')
cursor = banco.cursor()
def carrega_verbalizacoes(paramlista_verbalizacao):
    global cursor, banco, relation_verbalizations, relations, valid_conditions
    relations = ['no_relation',]
    cod_relacao = paramlista_verbalizacao
    valid_conditions = {}
    relation_verbalizations = {}
    sql = 'SELECT id, relation FROM TB_RELACAO WHERE id in '+cod_relacao
    cursor.execute(sql)
    resposta = cursor.fetchall()
    for n, relation in resposta:
        relations.append(relation)
```

```
        # verbalizacao
        sql2 = 'SELECT verbalizacao FROM TB_VERBALIZACAO WHERE id_relation
= '+str(n)+' AND verbalizacao IS NOT NULL'
        verba = cursor.execute(sql2)
        listaverbalizacao = []
        for v in verba:
            listaverbalizacao.append(v[0])
        relation_verbalizations[relation] = listaverbalizacao
        #validRelations
        sql3 = 'SELECT texto FROM TB_VALIDACAO WHERE id in (SELECT
DISTINCT id_validCondition FROM TB_VERBALIZACAO WHERE id_relation =
'+str(n)+' ORDER BY id_validCondition ASC)'
        cond = cursor.execute(sql3)
        listavalidRelations = []
        for c in cond:
            listavalidRelations.append(c[0])
        valid_conditions[relation] = listavalidRelations
    print('-'*10)
    print(relations)
    print(len(relations),u' Relações carregadas.')
    print('-'*10)
    print(relation_verbalizations)
    print(' Verbalizações carregadas referentes a '+str(len(relation_verbalizations))+'
relações.')
    print('-'*10)
    print(valid_conditions)
    print(len(valid_conditions), u' validConditions carregadas.')
def carrega_configuracoes():
    global clf
    print(u'4-Iniciando carregamento de configurações...')
    tempo_inicial = time.time()
    global relation_verbalizations, relations, valid_conditions, n_th, mod_er
    clf = NLIRelationClassifierWithMappingHead(
        labels=relations,
        template_mapping=relation_verbalizations,
        valid_conditions=valid_conditions,
        pretrained_model = mod_er,
        use_cuda=True,
        negative_threshold=n_th
    )
    print(u'  Carregamento finalizado.')
    tempo_final = time.time()
    tempo_execucao = tempo_final - tempo_inicial
    print('Tempo total: ',str(tempo_execucao))
    return clf
# função utilizada para carregar parte do corpus
def carrega_corpus(lista_id):
    global cursor, banco
```

```
    string_lista_id = ', '.join(map(str, lista_id))
    print(string_lista_id)
    print(type(string_lista_id))
    sql = 'SELECT id, sentenca, entity1, entity2, entity_type1, entity_type2, referencia
FROM '+paramExtracao+' WHERE id_corpus in ('+str(string_lista_id)+")"
    cursor.execute(sql)
    resultado2 = cursor.fetchall()
    corpus = []
    for id, sentenca, entity1, entity2, entity_type1, entity_type2, referencia in resultado2:
        sentenca = sentenca.replace("'","")
        if entity_type1 == 'B-PER' or entity_type1 == 'B-PESSOA':
            tipo1 = 'PERSON'
        elif entity_type1 == 'B-LOC' or entity_type1 == 'B-LOCAL':
            tipo1 = 'LOCAL'
        elif entity_type1 == 'B-ORG' or entity_type1 == 'B-ORGANIZACAO':
            tipo1 = 'ORGANIZATION'
        elif entity_type1 == 'B-TIM' or entity_type1 == 'B-TIME' or entity_type1 == 'B-
TEMPO':
            tipo1 = 'TIME'
        else:
            continue
        if entity_type2 == 'B-PER' or entity_type2 == 'B-PESSOA':
            tipo2 = 'PERSON'
        elif entity_type2 == 'B-LOC' or entity_type2 == 'B-LOCAL':
            tipo2 = 'LOCAL'
        elif entity_type2 == 'B-ORG' or entity_type2 == 'B-ORGANIZACAO':
            tipo2 = 'ORGANIZATION'
        elif entity_type2 == 'B-TIM' or entity_type2 == 'B-TIME' or entity_type2 == 'B-
TEMPO':
            tipo2 = 'TIME'
        else:
            continue
        corpus.append(REInputFeatures(subj=entity1, obj=entity2,
pair_type=tipo1+':'+tipo2, context=sentenca, label=str(id)))
    print(len(corpus), u' de corpus.')
    return(corpus)
def processa(corp, vTrue, vtopk) -> 'list':
    global clf
    resultado = clf.predict(corp, return_confidences=vTrue, topk=vtopk)
    return(resultado)
def executa(corpus, clf):
    tempo_inicial = time.time()
    global cursor, banco, xref, xserieid, xproc, mod_er, paramlista_len, paramExtracao
    if paramlista_len > 1:
        vtopk = 3
    else:
        vtopk = 2
    n = 0
```

```
    while True:
        if n >=len(corpus):break
        y = int(corpus[n].label)
        print('-'*20)
        corp = [corpus[n],]
        # essa segunda leitura é necessária porque o corpus preparado para extração não
contém todos os elementos necessários para o registro do resultado
        sql = 'SELECT id, sentenca, entity1, entity2, entity_type1, entity_type2, referencia,
id_corpus FROM '+paramExtracao+' WHERE id = '+str(y)
        cursor.execute(sql)
        result = cursor.fetchone()
        xid = result[0]
        xsentenca = result[1]
        xentity1 = result[2]
        xentity2 = result[3]
        xentity_type1 = result[4]
        xentity_type2 = result[5]
        xreferencia = xref+result[6]
        xid_corpus = result[7]
        xid_extracao = str(xserieid)+str(xid)
        print('xid_extracao: ',xid_extracao)
        print('xid_corpus: ',xid_corpus)
        sql4 = "SELECT id FROM TB_RESULTADO WHERE id_extracao
="+str(xid_extracao)
        cursor.execute(sql4)
        r = cursor.fetchone()
        if r is None:
            sql2= "INSERT INTO TB_RESULTADO (sentenca, entity1, entity2,
entity_type1, entity_type2, referencia, id_corpus, id_extracao) VALUES
('"+str(xsentenca)+"', '"+str(xentity1)+"', '"+str(xentity2)+"', '"+str(xentity_type1)+"',
'"+str(xentity_type2)+"', '"+str(xreferencia)+"', "+str(xid_corpus)+",
"+str(xid_extracao)+")"
            cursor.execute(sql2)
            resultado = processa(corp, True, vtopk)
            res_relation1 = resultado[0][0]
            res_relation1_desc = res_relation1[0]
            res_relation1_score = res_relation1[1]
            print(' ',res_relation1_score, res_relation1_desc)
            res_relation2 = resultado[0][1]
            res_relation2_desc = res_relation2[0]
            res_relation2_score = res_relation2[1]
            print(' ',res_relation2_score, res_relation2_desc)
            if vtopk == 3:
                res_relation3 = resultado[0][2]
                res_relation3_desc = res_relation3[0]
                res_relation3_score = res_relation3[1]
                print(' ',res_relation3_score, res_relation3_desc)
            if paramlista_len > 1:
```

```
            # mínimo de 3 verbalizações
            sql3 = "UPDATE TB_RESULTADO SET proc = '"+xproc+"', mod_er =
'"+mod_er+"',relation1 = '"+res_relation1_desc+"',score1 =
'"+str(res_relation1_score)+"',relation2 = '"+res_relation2_desc+"',score2 =
'"+str(res_relation2_score)+"',relation3 = '"+res_relation3_desc+"',score3 =
'"+str(res_relation3_score)+"' WHERE id_extracao="+str(xid_extracao)
        else:
            # mínimo de 2 verbalizações
            sql3 = "UPDATE TB_RESULTADO SET proc = '"+xproc+"', mod_er =
'"+mod_er+"',relation1 = '"+res_relation1_desc+"',score1 =
'"+str(res_relation1_score)+"',relation2 = '"+res_relation2_desc+"',score2 =
'"+str(res_relation2_score)+"' WHERE id_extracao="+str(xid_extracao)
        cursor.execute(sql3)
        banco.commit()
        print('-'*20)
      n += 1
  banco.close()
  tempo_final = time.time()
  tempo_execucao = tempo_final - tempo_inicial
  print('Tempo total: ',str(tempo_execucao))
def cria_conjunto_MinMax(idmin, idmax):
  sql = 'SELECT id FROM TB_CORPUS WHERE id >= '+str(idmin)+' AND id <=
'+str(idmax)
  cursor.execute(sql)
  docpre = cursor.fetchall()
  docs = []
  for d in docpre:
    docs.append(d[0])
  return(docs)
def cria_conjunto_verbalizacao(paramlista):
  cod_relacao = paramlista
  sql = 'SELECT DISTINCT id_corpus FROM TB_VERBALIZACAO WHERE
id_relation in '+cod_relacao
  cursor.execute(sql)
  docpre = cursor.fetchall()
  docs = []
  for d in docpre:
    docs.append(d[0])
  return(docs)
def adiciona_placebo(conjuntopre):
  for x in range(101,151):
    conjuntopre.append(x)
  return(conjuntopre)
def main():
  global paramlista
  print('-'*20)
  print(u'1-Carregando relações,verbalizações e relações válidas...')
  carrega_verbalizacoes(paramlista_verbalizacao)
```

```
    print('-'*20)
    clf = carrega_configuracoes()
    print('-'*20)
    # cria conjunto
    print(u'5-Criando conjunto...')
    conjuntopre = cria_conjunto_MinMax(paramConjuntoMin,paramConjuntoMax)
    if adiciona_placebo_flag == True:
        conjunto = adiciona_placebo(conjuntopre)
    else:
        conjunto = conjuntopre
    print('-'*20)
    print(u'6-Criando corpus...')
    print(len(conjunto))
    print('-'*20)
    print('-'*20)
    x = carrega_corpus(conjunto)
    print('-'*20)
    print(u'7-Executando...')
    executa(x, clf)
    print('-'*20)
    try:
        banco.close()
    except:
         pass
    print(u'8-Finalizado.')

if __name__ == '__main__':
    main()
```

APÊNDICE D

Código de programação: confusion_matrix.py

```
# Script para calcular métricas de avaliação
from sklearn.metrics import confusion_matrix
import sqlite3
import numpy as np
from datetime import datetime
now = datetime.now()
# Defina o número do processamento
vproc = '108'
# Defina o código da relação analisada para busca do universo positivo
cod_relacao = '(1,2,3,4,5,6,7,8)'   #cod_relacao = '(1,2,3,4,5,6,7,8)'
cod_relacao2 = {1,2,3,4,5,6,7,8}
# Defina o conjunto Universo
global universo
# colocar um número a mais no maxrange
universo = set(range(1, 151))
banco = sqlite3.connect('corpus.db')
cursor = banco.cursor()
print(u'AVALIAÇÃO DO PROCESSAMENTO')
print('-'*20)
print('Avaliação do proc = ',vproc)
print(u'Relações: ', cod_relacao)
print('Data de Processamento: ', now)
print('-'*20)
# Cálculo do gabarito positivo (relação verdadeira)
sql2 = """
   SELECT DISTINCT id_corpus, id_relation
   FROM TB_VERBALIZACAO V
   LEFT JOIN TB_RELACAO  R ON V.id_relation = R.id
   WHERE id_corpus <= ?
   ORDER BY id_corpus
"""
cursor.execute(sql2, (max(universo),))
resposta2 = cursor.fetchall()
positivo = set((id_corpus, id_relation) for id_corpus, id_relation in resposta2 if
id_relation in cod_relacao2)
positivo3 = set((id_corpus) for id_corpus, id_relation in resposta2 if id_relation in
cod_relacao2)
print()
print('id_corpus positivos: ', len(positivo3))
print(sorted(positivo3))
print()
print('gabarito positivo: ', len(positivo))
print(sorted(positivo))
# Cálculo do resultado
```

```
sql1 = """
SELECT
      id_corpus,
      id_relation
   FROM (
      SELECT
         id_corpus,
         relation1, REL.id as id_relation,
         MAX(score1) AS max_score
      FROM TB_RESULTADO RES
      LEFT JOIN TB_RELACAO REL ON RES.relation1 = REL.relation
      WHERE proc = """+vproc+""" AND relation1 <> 'no_relation' AND id_corpus <=
?
      GROUP BY id_corpus
   ) AS max_scores
   WHERE max_scores.max_score IS NOT NULL
"""
cursor.execute(sql1, (max(universo),))
resposta1 = cursor.fetchall()
resultado = set((id_corpus, id_relation) for id_corpus, id_relation in resposta1)
resultado3 = set((id_corpus) for id_corpus, id_relation in resposta1)
print()
print('id_corpus resultado: ', len(resultado3))
print(resultado3)
print()
print('resultado: ', len(resultado))
print(sorted(resultado))
global VP, FP, VN, FN, gVP, gFP, gVN, gFN
VP = 0
FP = 0
VN = 0
FN = 0
gVP = []
gFP = []
gVN = []
gFN = []
def proc_positivo(xres, xpositivo):
   for pos in xpositivo:
      if pos == xres:
         return True
   return False
def avalia_resultado(vresultado, vpositivo):
   global VP, FP, gVP, gFP
   for res in vresultado:
      if proc_positivo(res,vpositivo) == True:
         VP += 1
         gVP.append(res)
      else:
```

```
            FP += 1
            gFP.append(res)
    return
def avalia_negativos(xresultado3, xpositivo3, xuniverso):
    global VN, FN, gVN, gFN
    for uni in xuniverso:
        if uni not in xresultado3:
            if uni not in xpositivo3:
                VN += 1
                gVN.append(uni)
            else:
                FN += 1
                gFN.append(uni)
    return
avalia_resultado(sorted(resultado), sorted(positivo))
avalia_negativos(resultado3, positivo3, universo)
print('Avaliação do proc = ',vproc)
print(u'Relações: ', cod_relacao)
print()
print(u'          MATRIZ DE CONFUSÃO          ')
print()
print('VP: ',VP)
print('FP: ',FP)
print('VN: ',VN)
print('FN: ',FN)
try:
    vPrecision = VP / (VP + FP)
except:
    vPrecision = 'NA'
print('vPrecision: ', vPrecision)
print()
vRecall = VP / (VP + FN)
print('vRecall/vSensitivity: ', vRecall)
print()
try:
    vF1Score = 2*(vPrecision*vRecall) / (vPrecision + vRecall)
except:
    vF1Score = 'NA'
print('vF1Score: ', vF1Score)
print()
vAccuracy = (VP + VN) / (VN + VP + FP + FN)
print('vAccuracy: ', vAccuracy)
```