Trabalho de Conclusão de Curso

Reconhecimento e análise de entidades nomeadas em textos não-estruturados usando LLMs e redes complexas

de **Yuri Dimitri Ramos Costa**

orientado por
Prof. Dr. Thales Miranda de Almeida Vieira

Universidade Federal de Alagoas
Instituto de Computação
Maceió, Alagoas
23 de Outubro de 2024

UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação

RECONHECIMENTO E ANÁLISE DE ENTIDADES NOMEADAS EM TEXTOS NÃO-ESTRUTURADOS USANDO LLMS E REDES COMPLEXAS

Trabalho de Conclusão de Curso submetido ao Instituto de Computação da Universidade Federal de Alagoas como requisito parcial para a obtenção do grau de Engenheiro de Computação.

Yuri Dimitri Ramos Costa

Orientador: Prof. Dr. Thales Miranda de Almeida Vieira

Banca Avaliadora:

Marcelo Costa Oliveira	Prof. Dr., IC-UFAL
Adriano Oliveira Barbosa	Prof. Dr., UFGD

Maceió, Alagoas
23 de Outubro de 2024

Catalogação na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária: Maria Helena Mendes Lessa – CRB-4 – 1616

C837r Costa, Yuri Dimitri Ramos.
Reconhecimento e análise de entidades nomeadas em textos não-estruturados usando LLMs e redes complexas / Yuri Dimitri Ramos Costa. – Maceió, 2024.
62 f. : il., grafs. e tabs. color.

Orientador: Thales Miranda de Almeida Vieira.
Monografia (Trabalho de conclusão de curso em Engenharia de Computação) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2024.

Bibliografia: f. 57-62.

1. Processamento de linguagem natural (Computação). 2. Redes complexas. 3. Transformers. I. Título.

CDU: 004.41

UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação

RECONHECIMENTO E ANÁLISE DE ENTIDADES NOMEADAS EM TEXTOS NÃO-ESTRUTURADOS USANDO LLMS E REDES COMPLEXAS

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação pelo Instituto de Computação da Universidade Federal de Alagoas, aprovada pela comissão examinadora que abaixo assina.

gov.br Documento assinado digitalmente
THALES MIRANDA DE ALMEIDA VIEIRA
Data: 23/10/2024 14:57:29-0300
Verifique em https://validar.iti.gov.br

Thales Miranda de Almeida Vieira,
Prof. Dr., IC-UFAL, Orientador

gov.br Documento assinado digitalmente
MARCELO COSTA OLIVEIRA
Data: 24/10/2024 13:43:37-0300
Verifique em https://validar.iti.gov.br

Marcelo Costa Oliveira,
Prof. Dr., IC-UFAL, Examinador

gov.br Documento assinado digitalmente
ADRIANO OLIVEIRA BARBOSA
Data: 23/10/2024 15:06:06-0300
Verifique em https://validar.iti.gov.br

Adriano Oliveira Barbosa,
Prof. Dr., UFGD, Examinador

Dedicatória

Dedico este trabalho aos meus pais, Magna e Lenin, pelo apoio, pelos ensinamentos e por acreditarem em mim em todos os momentos.

Agradecimentos

À Deus, por ter me guiado, me protegido e me capacitado em meio aos desafios.

À minha mãe, Magna, por todo o apoio e incetivo nos momentos de dificuldades e ao meu pai, Lenin, pelos ensinamentos ao longo da minha vida.

À minha família, por me fornecerem apoio e compreensão em cada etapa desta jornada.

Aos amigos que estiveram comigo, pelo incetivo, pelas conversas e pelos momentos de descontração que tornaram o caminho mais leve.

Ao meu orientador, Thales, por todo o apoio e colaboração desde a iniciação científica, contribuindo para minha formação acadêmica e profissional.

À Universidade Federal de Alagoas, à banca examinadora e todos os professores que fizeram parte da minha trajetória, por fornecerem uma educação de qualidade que me permitiu e permite muitos outros a alçarem sonhos maiores.

À todos aqueles que, de alguma forma, contribuíram para a realização deste sonho.

Yuri Dimitri Ramos Costa

"A educação é a arma mais poderosa que você pode usar para mudar o mundo."

Nelson Mandela

Resumo

A disponibilidade de dados textuais na internet vem crescendo exponencialmente. Se o acesso a informação já foi um grande desafio em outros momentos, hoje, o grande volume de conteúdo disponível exige a utilização de métodos cada vez mais sofisticados para automatizar a extração de informações. Seja por meio de livros, artigos, e-mails ou postagens em redes sociais, as informações disponíveis nesses meios podem contribuir para a geração de ideias inovadoras, a melhor compreensão do público alvo em campanhas publicitárias e o avanço da ciência por meio de revisões de literatura. Dessa forma, os dados textuais tem se mostrado valiosos em um mundo cada vez mais conectado.

O reconhecimento de entidade nomeadas (NER) permite identificar palavras de um texto que se referem a um tema em comum. A estruturação das entidades nomeadas em um grafo de relações fornece uma estrutura capaz de representar o contexto do conjunto de dados, permitindo a identificação de padrões.

Com os recentes avanços dos modelos generativos *Large Language Models* (LLM), o presente trabalho avaliou o desempenho desse tipo de modelo no reconhecimento de entidades nomeadas, discutindo acerca de vantagens e desvantagens por meio de uma análise quantitativa da tarefa realizada, comparando o ChatGPT com modelos como BERT e BiLSTM. Além disso, uma análise qualitativa busca avaliar as capacidades dos modelos BERT e ChatGPT na geração de grafos de relações entre entidades nomeadas, além de propor técnicas de exploração para esse tipo de rede complexa.

Desta forma, buscou-se neste estudo apresentar uma metodologia para a geração de representações em redes complexas de dados textuais. A metodologia apresentada permite duas abordagens: a utilização de modelos LLM, como o ChatGPT, por meio da engenharia de *prompt*, necessitando uma quantidade mínima de exemplos, e o treinamento ou refino de modelos estado da arte, como o BERT.

Palavras-chave*: *Processamento de linguagem natural*; *Redes complexas*; *Transformers

Abstract

The availability of textual data on the internet has been growing exponentially. While access to information was a major challenge in the past, today, the vast amount of available content requires increasingly sophisticated methods to automate information extraction. Whether through books, articles, emails, or social media posts, the information available in these media can contribute to generating innovative ideas, better understanding the target audience in advertising campaigns, and advancing science through literature reviews. In this way, textual data has proven to be valuable in an increasingly connected world.

Named entity recognition (NER) allows identifying words in a text that refer to a common theme. Structuring the named entities in a relation graph provides a framework capable of representing the context of the dataset, enabling the identification of patterns.

With the recent advances in generative models, such as Large Language Models (LLM), this study evaluated the performance of this type of model in named entity recognition, discussing the advantages and disadvantages through a quantitative analysis of the task performed, comparing ChatGPT with models like BERT and BiLSTM. Additionally, a qualitative analysis seeks to evaluate the capabilities of the BERT and ChatGPT models in generating relation graphs between named entities, as well as proposing exploration techniques for this type of complex network.

Thus, the aim of this study was to present a methodology for generating representations in complex networks of textual data. The presented methodology allows two approaches: the use of LLM models, such as ChatGPT, through prompt engineering, requiring a minimal number of examples, and the training or fine-tuning of state-of-the-art models, such as BERT.

***Keywords*: *Natural language processing*; *Complex networks*; *Transformer models*.**

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

NN *Neural Networks*

RNN *Recurrent Neural Networks*

NER *Named Entity Recognition*

GPT *Generative Pre-trained Transformers*

BERT *Bidirecional Encoder Representations from Transformers*

LLM *Large Language Models*

MLM *Masked Language Model*

Sumário

Capítulo 1

Introdução

Todos os dias a internet produz uma grande quantidade de dados textuais, em sua maioria de maneira não estruturada e escrita em linguagem natural. Segundo Hirschberg and Manning (2015), as linguagens naturais, ou linguagens humanas, utilizadas no dia a dia, como português e inglês, possuem ambiguidades e muito da compreensão é feita utilizando do contexto e do repertório cultural. Por tais características, essas linguagens são difíceis de serem processadas por meio de instruções lógico matemáticas, como é feito nos computadores por meio de linguagens formais, argumenta o autor. Para extrair informações a partir desse tipo de dados, surgiu o campo do processamento de linguagem natural dentro do contexto da inteligência artificial.

O reconhecimento de entidades nomeadas (NER) é uma forma de extração de informação de dados textuais não-estruturados que visa reconhecer palavras de um texto que se referem a um tema em comum, ou instâncias de alguns "tipos de expressão", conforme sugere Chinchor and Robinson (1998). Por exemplo, podemos ter uma entidade para localizações e desejamos obter todas as palavras do texto que se referem a uma localização. Nesse caso, a palavra Brasil seria uma entidade nomeada de localização. No entanto, a depender das entidades a qual se deseja identificar, a identificação correta das palavras pode depender mais ou menos do contexto.

Embora o NER seja capaz de identificar elementos chaves em um texto, a representação das entidades nomeadas por meio de um grafo de relações adiciona a análise uma camada de informação estruturada que pode ser valiosa por permitir a identificação da frequência a qual cada entidade nomeada está associada e a relevância de uma entidade nomeada em relação a outra, por exemplo. Assim, a análise da representação em grafos enriquece a metodologia apresentada, trazendo informações sobre a centralidade e a influência de cada entidade dentro do conjunto de dados. Além disso, a técnica de detecção de comunidades buscou agrupar entidades nomeadas em temas.

Dentre os objetivos, o presente trabalho visa realizar o reconhecimento de entidades nomeadas para a estruturação de informações em redes complexas extraídas automaticamente de dados textuais. Para o desenvolvimento e avaliação da metodologia foi utilizado

um conjunto de dados composto por postagens da rede social Stack Exchange[1]. Os tipos de entidades nomeadas escolhidos estão relacionadas à países, orgãos públicos, empresas, pessoas, cargos, localizações e elementos do direito como leis e jurisprudências.

Além disso, foi realizada uma exploração do grafo de relações entre as entidades nomeadas com o objetivo de avaliar o uso de ferramentas como a análise de centralidade de nós, a identificação de subgrafos relevantes e a detecção de comunidades para extrair informações a partir de padrões em toda a rede social. Dessa forma, essa etapa busca a compreensão do conjunto de dados de maneira mais completa.

A geração de redes complexas utilizando reconhecimento de entidades nomeadas tem sido utilizada em diferentes contextos. Sawicki et al. (2024) busca formar um grafo estabelecendo similaridades entre comunidades da rede social Reddit[2] baseado nos tópicos comuns identificados utilizando NER. Além disso, o método proposto por Wei et al. (2023) utiliza o reconhecimento de entidades nomeadas em artigos publicados para construir grafos de conhecimento. Um método baseado em engenharia de *prompt* para reconhecimento de entidades nomeadas em LLMs é proposto por Wang et al. (2023). Assim, essas abordagens demonstram o potencial do NER para estruturar informações complexas acerca de diferentes domínios.

1.1 Justificativa

As redes sociais são uma fonte aberta e rica de dados contendo informações e opiniões acerca de diferentes temas. Dados coletados de redes sociais podem ser utilizados para obter informações valiosas para organizações de ajuda humanitária, identificando problemas sérios quando ocorrem e quem está sendo afetado, conforme exemplifica Hirschberg and Manning (2015). Devido a sua popularidade, as redes sociais possuem uma grande diversidade de pensamentos e podem indicar novas tendências mundiais. Por outro lado, devido a dificuldade de monitoramento do conteúdo nelas presente também se tornaram meios de disseminação de desinformação em massa.

Conforme Talagala (2022), apesar da frase "Os dados são o novo petróleo" enfatizar o valor dos dados, assim como o petróleo, eles precisam ser refinados para se tornarem úteis. As redes sociais são um grande repositório de informações mais próximas aos usuários, com a análise do conteúdo textual desses meios de comunicação podendo contribuir para retenção de consumidores, criação de novos modelos de negócios e aumento da eficiência na publicidade.

A extração de informações desses meios de comunicação esbarram no desafio do grande volume de dados que tornou a análise manual desse conteúdo uma tarefa inviável e com isso surgiu a necessidade de métodos automatizados para extrair informações. Para da-

[1] `https://politics.stackexchange.com/`
[2] `https://www.reddit.com/`

dos textuais, métodos de reconhecimento de entidades nomeadas podem ser utilizados estabelecer relações entre os temas em maior evidência dentro de redes sociais de modo a identificar tendências dentre os usuários levando em consideração o contexto ao qual as palavras são utilizadas e não apenas os termos mais frequentes.

Esse trabalho busca desenvolver um método para a estruturação de informações a partir de um conjunto de dados textuais, buscando minimizar a quantidade de trabalho manual por meio de *Large Language Models*. Ainda que a experimentação tenha sido realizada por meio da aplicação do método a um conjunto de dados sobre o cenário político mundial obtido a partir das interações textuais na rede social *Stack Exchange*, o método pode ser replicado para outros tipos de conjuntos, como análise de documentos legais.

1.2 Objetivos

1.2.1 Objetivos Gerais

O presente trabalho visa desenvolver um método para extração de informações por meio do reconhecimento de entidades nomeadas e estruturação em rede complexa. Busca ainda avaliar a viabilidade do método em obter informações acerca do cenário político mundial a partir da aplicação em um conjunto de dados obtido de comentários de usuários da rede social *Stack Exchange* acerca do tema. Além disso, busca realizar comparação do desempenho na tarefa de reconhecimento de entidades nomeadas dos modelos de *Large Language Model* (LLM) BERT e *ChatGPT*, utilizando a arquitetura de rede neural BiLSTM enquanto modelo baseline para comparação.

1.2.2 Objetivos Específicos

- Estudar técnicas para o reconhecimento de entidades nomeadas;
- Desenvolver um método capaz de extrair informações de interesse e estruturar em formato de grafo utilizando aprendizagem de máquina;
- Avaliar o uso de engenharia de *prompt* em *Large Language Models* para realização do reconhecimento de entidades nomeadas;
- Avaliar o desempenho dos modelos ChatGPT e BERT na tarefa de reconhecimento de entidades nomeadas.

1.3 Organização do Trabalho

O trabalho está organizado em cinco capítulos principais. A fundamentação teórica (capítulo 2) apresenta os conceitos e teorias que serviram como base para a pesquisa. Em seguida, o capítulo trabalhos relacionados (capítulo 3) busca contextualizar os avanços na área do reconhecimento de entidades e as aplicações da tarefa na geração de grafos. Posteriormente, o capítulo 4 apresenta a metodologia para realização do reconhecimento de entidades nomeadas e estruturação em grafos, descrevendo também as abordagens utilizadas para extração de informações. No capítulo de experimentos (capítulo 5), são apresentados o conjunto de dados utilizado e os resultados obtidos, juntamente com a discussão acerca da viabilidade do método. Por fim, a conclusão resume as principais contribuições do estudo e sugere direções para pesquisas futuras.

Capítulo 2

Fundamentação teórica

2.1 Processamento de linguagem natural

Conforme Hirschberg and Manning (2015), o processamento de linguagem natural (PLN), também conhecido como linguística computacional, é o subcampo da ciência da computação que trata da utilização de métodos computacionais com o próposito de aprender, entender e produzir conteúdo em linguagem humana.

O objetivo da ciência linguística é poder caracterizar e explicar a multidão de observações linguísticas circulando entre nós em conversas, escrita e outras mídias (Manning and Schütze, 1999). No entanto, o autor menciona que, para linguagens naturais (ou linguagens humanas), simplesmente não é possível prover uma caracterização exata e completa de enunciados bem formados que os divida claramente de todas as outras sequências de palavras, que são consideradas expressões mal formadas, pois as pessoas estão sempre reajustando a forma de se comunicar conforme suas próprias necessidades. Assim, tendo em vista que uma caracterização formal não é possível, surgiu a necessidade de métodos estatísticos para que as máquinas compreendam melhor os padrões mais comuns do uso da linguagem.

2.1.1 Distância de Levenshtein

A distância de Levenshtein é uma métrica para comparar a diferença entre sequências de caracteres. O valor é obtido por meio da quantidade de operações aplicadas a caracteres individuais necessárias para que uma sequência A fique igual a uma sequência B. As operações de caracteres possíveis podem ser inserção, deleção ou substituição.

Sequência 1	**Sequência 2**	**Distância**
alegria	alergia	2
felicidade	Felicidade	1

Tabela 2.1: Exemplos de distâncias de Levenshtein.

Uma outra maneira de comparar sequências de caracteres é por meio do algoritmo de Ratcliff/Obershelp, que consiste no número de caracteres iguais dividido pelo número total de caracteres nas duas sequências.

2.2 Aprendizagem de máquina

A aprendizagem de máquina, ou *Machine Learning*, é o subcampo da ciência da computação que busca algoritmos para que os computadores possam aprender automaticamente relações significativas e padrões a partir de exemplos e observações (Bishop, 2006).

Uma definição formal dada por Mitchell (1997) é que diz-se que um programa de computador aprende a partir de uma experência E relativo a uma classe de tarefas T e medida de perfomance P, se a sua perfomance em tarefas da classe T, medidas por P, melhoram com a experiência E, em tradução livre. O autor exemplifica dizendo que um programa que aprende a jogar xadrez deve melhorar sua perfomance medida por sua habilidade de ganhar na classe de tarefas envolvendo jogar xadrez a partir da experiência obtida jogando partidas contra ele mesmo.

A literatura costuma separar os algoritmos de aprendizagem de máquina em três principais categorias, algoritmos de aprendizado supervisionado, aprendizado não supervisionado e aprendizagem por reforço, ainda que variações dos paradigmas de aprendizado são encontradas, como é o caso do aprendizado semi-supervisionado, onde alguns exemplos do conjunto de treinamento possuem os valores de saída esperados, mas outros não (Goodfellow et al., 2016).

2.2.1 Aprendizado supervisionado

Aplicações as quais os dados de treinamento contém exemplos do vetor de entrada com seus vetores objetivo correspondente são conhecido como problemas de aprendizado supervisionado (Bishop, 2006). Desse modo, quando durante o treinamento é fornecido ao algoritmo o vetor de propriedades como entrada e também o vetor de saída esperado, dizemos que se trata de um algoritmo de aprendizado supervisionado.

Quando o vetor de saída esperado corresponde a valores discretos, chamamos de problemas de classificação. Quando o vetor de saída esperado contém variáveis contínuas, então chamamos de problemas de regressão.

2.2.2 Aprendizado não supervisonado

Algoritmos de aprendizado não supervisionado recebem um vetor de entrada sem a informação dos vetores objetivo. O objetivo desses algoritmos é aprender propriedades úteis acerca do conjunto de dados (Goodfellow et al., 2016). Dentre as tarefas mais comuns, o autor menciona o agrupamento (*clustering*), que consiste em dividir o conjunto de dados em grupos, ou aglomerações, contendo exemplos com propriedades similares.

Aprendizado generativo

Segundo Zorzi et al. (2013), o aprendizado generativo tem por objetivo modelar a distribuição conjunta $P(X, Y)$ de variáveis observadas e latentes normalmente por meio da maximização de probabilidades acerca dos dados observados. Ou seja, podemos dizer que esse tipo de aprendizado busca aprender a representar a distribuição dos dados, permitindo, por exemplo, a geração de novos exemplos com base na distribuição aprendida.

Modelagem de linguagem e geração de textos

O objetivo da modelagem de linguagem é aprender a função de probabilidade conjunta de sequências de palavras em uma linguagem (Bengio et al., 2003). A abordagem mais comum para geração de textos é feita fornecendo uma sequência de números inteiros de entrada (texto dividido em tokens) a um algoritmo capaz de produzir um vetor n-dimensional de probabilidades para a próxima palavra do texto, onde n é o número de palavras no vocabulário dos textos de treinamento.

2.2.3 Métricas de avaliação

Foram desenvolvidas uma grande variedade de métricas com base em cálculos matemáticos para avaliar a perfomance dos modelos de aprendizagem de máquina. Por meio dessas métricas é possível estimar o tipo de erro que o modelo realiza e assim decidir sobre o quão adequado é a sua utilização para uma determinada tarefa. Dentre as métricas para a avaliação da perfomance mais utilizadas estão a acurácia, a precisão e a cobertura.

Acurácia

De maneira intuitiva, a acurácia é uma métrica de classificação que indica a porcentagem de predições acertadas dentro o total de predições.

$$\text{Acurácia} = \frac{\text{predições corretas}}{\text{total de predições}} \tag{2.1}$$

Também pode ser definida da seguinte maneira:

$$\text{Acurácia} = \frac{TP + TN}{\text{TP+TN+FP+FN}} \tag{2.2}$$

De modo que TP é a quantidade de predições verdadeiro positivos, TN é a quantidade de predições verdadeiro negativos, FP é a quantidade de predições falso positivos e FN é a quantidade de predições falso negativos. Um verdadeiro positivo ocorre quando o valor esperado era positivo e a predição foi de fato positivo. Um falso positivo ocorre quando o valor esperado era negativo e a predição foi positivo.

Precisão

A precisão é uma medida que indica uma porcentagem de quantas predições positivas feitas pelo modelo foram de fato corretas, ou seja, quanto maior o valor indica uma menor quantidade de falsos positivos. A precisão está definida da seguinte forma:

$$\text{Precisão} = \frac{TP}{\text{TP+FP}} \tag{2.3}$$

Cobertura

A cobertura, ou *recall*, é uma medida que indica quantos dos exemplos da classe positiva foram identificados corretamente pelo modelo, ou seja, quanto maior indica uma menor quantidade de falsos negativos. Está definida da seguinte forma:

$$\text{Cobertura} = \frac{TP}{\text{TP+FN}} \tag{2.4}$$

F_1 score

A métrica F_1 estabelece uma relação entre as métricas de precisão e cobertura tendo como característica se tornar pior quanto maior a diferença entre precisão e cobertura.

$$F_1 \text{ score} = \frac{2 * \text{Precisão} * \text{Cobertura}}{\text{Precisão + Cobertura}} \tag{2.5}$$

2.2.4 Redes neurais artificiais

As redes neurais fornecem uma abordagem robusta para a aproximação de funções alvo com valores reais, discretos ou vetoriais. Elas consistem em um conjunto denso de unidades interconectadas onde cada unidade recebe um número de valor real como entrada (que pode ser a combinação da saída de outras unidades) e produzem uma única saída de valor real (que também pode servir de entrada para várias outras unidades). Por meio de algoritmos como *Backpropagation* (retropropagação) o método ajusta parâmetros do

modelo para melhor adequar-se aos pares de entrada e saída do conjunto de treinamento (Mitchell, 1997).

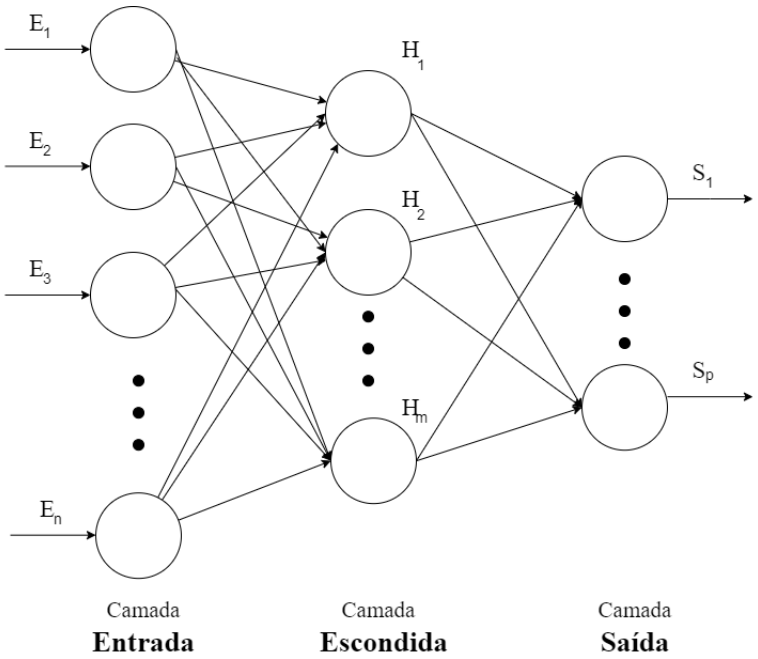

Figura 2.1: Representação visual de uma rede neural direta (Fonte: autor).

A figura 2.1 mostra uma rede neural direta onde E_1 até E_n são valores reais de entrada e S_1 até S_p são valores reais de saída. As arestas representam números reais que atuam como pesos a serem multiplicados pelos valores de entrada e os círculos representam as unidades neurais, que são aplicações de funções de ativação sobre a combinação dos valores reais passados pelas arestas. Para realizar a combinação vetorial a ser passada como entrada para a função de ativação é atribuido a cada aresta um peso (ou parâmetro) que é múltiplicado pelo valor de saída da unidade anterior e em seguida os valores obtidos são somados.

O aprendizado de uma rede neural é feito por meio do ajuste dos pesos para que dado uma entrada E, produza valores o mais próximos possível da saída esperada S. A medida de perfomance utilizada nesse algoritmo é a função de perda, que consiste em uma relação entre o valor de saída esperado e o valor de saída obtido após o ajuste dos pesos.

As funções de ativação calculam um valor real de saída baseado no valor de combinação de entrada. Dentre as funções de ativação mais utilizadas estão a função sigmoide, função linear retificada e tangente hiperbólica. O objetivo de aplicar os valores obtidos da combinação nessas funções é de tornar o modelo não-linear.

2.2.5 *Word embeddings*

Word embeddings são representações vetoriais para palavras que buscam estabelecer similaridades entre palavras, podendo representar relações semânticas e sintáticas por meio

de operações em um espaço vetorial. Como exemplo, em uma determinada representação, palavras sinônimas podem possuir representações vetoriais com menor distância euclidiana.

Mikolov et al. (2013) demonstra que palavras podem ter vários graus de similaridade e que as representações vetoriais podem representar relações complexas como o vetor para a palavra "Rei" subtraído do vetor da palavra "Homem" e somado ao vetor da palavra "Mulher" resultar em um vetor próximo da palavra "Rainha".

Dentre as principais formas de obter representações vetoriais para palavras é utilizar as representações disponibilizadas de modelos pré-treinados, como o GloVe (Pennington et al., 2014), ou adicionar ao modelo de rede neural uma camada responsável por aprender a melhor representação vetorial para as palavras presentes conjunto de dados.

2.2.6 Redes neurais recorrentes

As redes neurais recorrentes são um grupo de arquiteturas de redes neurais que permitem que uma entrada leve em consideração entradas anteriores em seu processamento atual. Conforme Hochreiter and Schmidhuber (1997a), redes neurais recorrentes podem, à princípio, usar conexões de *feedback* para armazenar representações de uma entrada recente em forma de ativações.

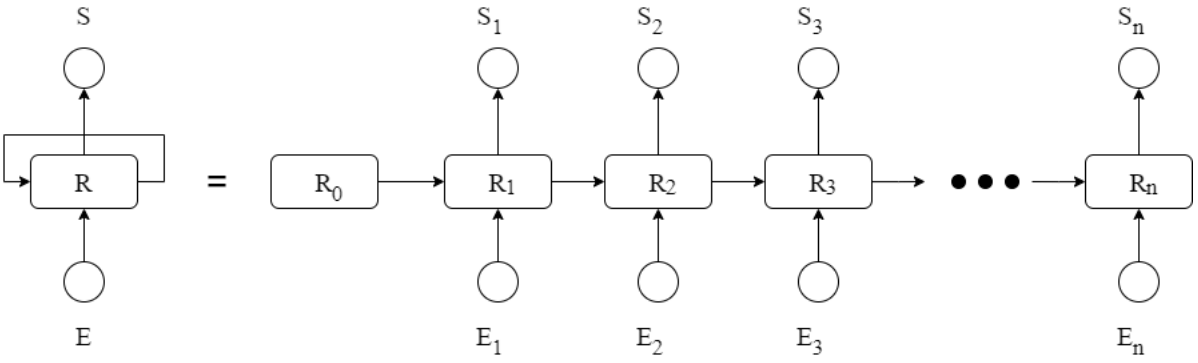

Figura 2.2: Representação visual de uma unidade recorrente de rede neural para n *timestamps* (Fonte: autor).

A figura 2.2 representa uma unidade de rede neural recorrente. A unidade recorrente recebe um valor de entrada E e utiliza o resultado produzido por ela mesma anteriormente para, combinado a entrada E, produzir uma saída S. Na lado direito da figura, em seu eixo horizontal, estão representadas os diferentes valores que uma mesma entrada E assume nos diferentes *timestamps* e repassa para a unidade recorrente. Ao centro do eixo vertical está representado uma unidade recorrente R recebendo informações da camada de entrada E e também do estado anterior em que ela se encontrava, produzindo uma saída para cada *timestamp*.

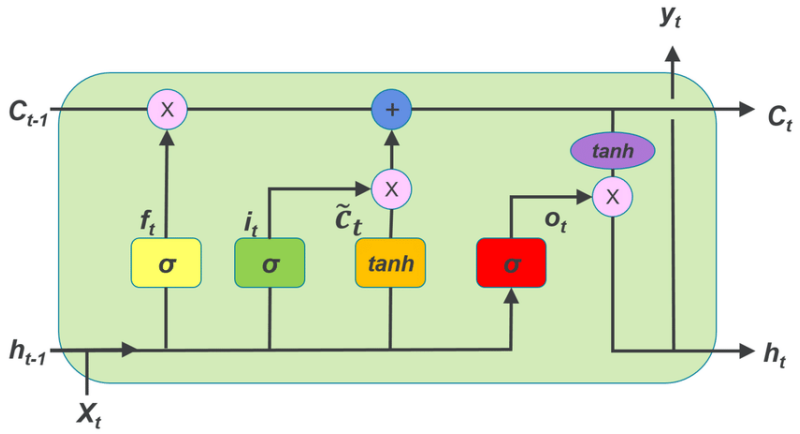

Figura 2.3: Representação visual da arquitetura de uma unidade LSTM (Shenfield and Howarth, 2020).

Hochreiter and Schmidhuber (1997b) introduziram o modelo de rede neural recorrente denominado *Long Short-Term Memory* (LSTM) cujo objetivo é prover a rede neural a capacidade de aprender a armazenar as informações mais relevantes por longos intervalos de tempo. Observando a figura 2.3 é possível observar que além de uma célula receber o seu estado anterior (h_{t-1}) e a entrada no tempo atual (x_t), passa a receber um estado da célula (c_{t-1}) que funciona como uma memória para armazenamento de informações importantes. Internamente, a célula permite o ajuste das informações passadas pela memória (c_{t-1}) por meio do ajuste dos parâmetros f_t e i_t. O parâmetro f_t ajusta a remoção ou manutenção de valores da memória, enquanto o parâmetro i_t permite a adição de valores a memória. O parâmetro O_t é responsável por permitir o ajuste do estado da memória.

2.2.7 Mecanismos de atenção

Os mecanismos de atenção foram propostos inicialmente por Bahdanau et al. (2014) para desempenhar a tarefa de tradução por máquina. O objetivo é permitir que os modelos relacionem uma palavra da entrada (ou vetor) as outras palavras da entrada.

As redes recorrentes anteriores possuiam uma grande dificuldade em acessar informações de pontos muito antigos da sequência de entrada. Isso foi tratado por meio da atribuição de pesos para cada parte da sequência (*tokens*) em relação as outras partes, formando uma matriz de atenção. Desse modo, para um determinado *token* x, pode haver um *token* y da sequência com maior peso para o *token* x sem necessariamente estar próximo na ordem da sequência. Posteriormente, o objetivo das camadas de atenção foram permitir que as redes aprendam quais são as partes específicas consideradas mais relevantes, ao invés de considerar para um *token* i somente os passos anteriores da sequência de entrada ($[0, i]$).

A ideia surge do problema do desalinhamento entre palavras durante uma tradução. Por exemplo, a tradução de *black cat* para o português é *gato preto*. Porém, podemos

observar que a ordem das palavras mudou durante a tradução. Ou seja, durante o primeiro passo recorrente, para realizar a tradução do texto *black*, a rede não possui a informação da palavra *cat*, que deve ter sua tradução como palavra inicial da sequência de saída.

Outro ponto importante é que as arquiteturas utilizadas anteriormente permitiam somente como entrada sentenças de tamanho fixo. Para as arquiteturas com entrada de tamanho fixo, quando a representação vetorial da sentença não atinge o tamanho máximo, a sequência é completada com valores de preenchimento pré-determinados. Para Bahdanau et al. (2014), no entanto, o maior problema é para sentenças de tamanho maior que o tamanho máximo. Nesses casos, é necessário comprimir toda a informação necessária ou perder informação. Quanto maior o tamanho máximo, pior se torna a perfomance para os modelos recorrentes anteriores. Os mecanismos de atenção tornam-se um método utilizado para modelos com sequências de entrada de tamanho indefinido.

2.2.8 Transformers

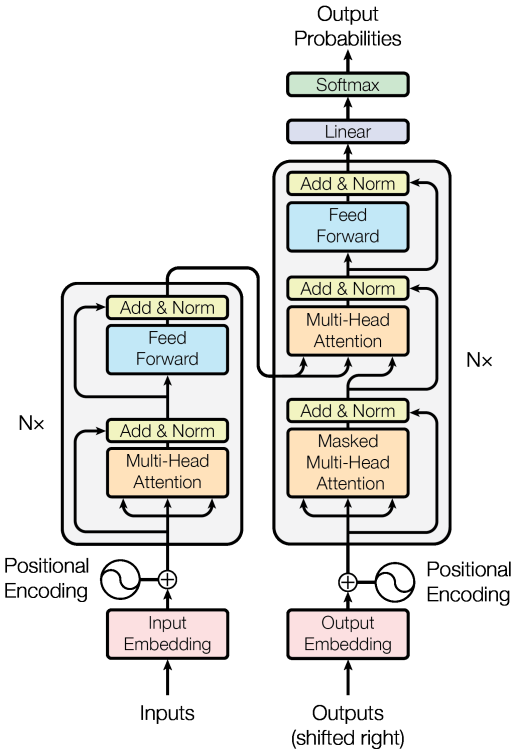

Figura 2.4: Representação visual da arquitetura *Transformer* (Vaswani et al., 2017).

Vaswani et al. (2017) propôe uma arquitetura de rede neural baseada unicamente em mecanismos de atenção, dispensando a necessidade de recorrências e convoluções. Essa arquitetura ficou conhecida como *Transformers*. A arquitetura se baseia no modelo *encoder-decoder*. Na figura 2.4 é possível observar na parte esquerda a representação do encoder que é ligado ao decoder na parte direita.

Na arquitetura *Transformer*, a entrada é fornecida inicialmente a camada de *Input*

Embedding, que busca particionar a entrada em *tokens*, que são representados por vetores que devem englobar, por meio dos seus valores, propriedades acerca daquela parte da entrada (*embeddings*). Esse vetor de *embedding* é aprendido durante o treinamento. Quando a entrada é um texto, os *tokens* gerados são normalmente palavras, porém em algumas arquiteturas podem ser pedaços de uma palavra. *Transformers* podem ser utilizados para outros tipos de entrada, como imagens, vídeos e áudios.

Em uma rede recorrente, cada *token* é fornecido para ser processado pela rede em um timestamp, porém o modelo *transformer* processa toda a entrada de uma única vez. A camada de *Positional Encoding* é responsável por codificar a informação de posição dos *tokens* no vetor de entrada. Após as camadas de atenção, normalização e rede *feed forward* o estado final do *encoder* é um vetor que representa a entrada. Esse vetor é fornecido como entrada para a camada de atenção do *decoder*. Essa informação contextual aprendida pelo encoder é utilizada junto as saídas produzidas pelo *decoder* para produzir novas saídas até que seja produzido um *token* final.

Dentre as vantagens dessa arquitetura, o autor menciona a quantidade de computação que pode ser paralelizada e o total de complexidade computacional por camada. Desse modo, o método proposto requer um número menor de operações quando comparado com as camadas recorrentes propostas anteriormente.

A partir da arquitetura de *Transformers*, dois modelos ganharam maior destaque: *Generative Pre-trained Transformers* (GPT) (Radford and Narasimhan, 2018) e *Bidirectional Encoder Representations from Transformers* (BERT) (Devlin et al., 2019).

2.2.9 *Generative Pre-trained Transformers* (GPT)

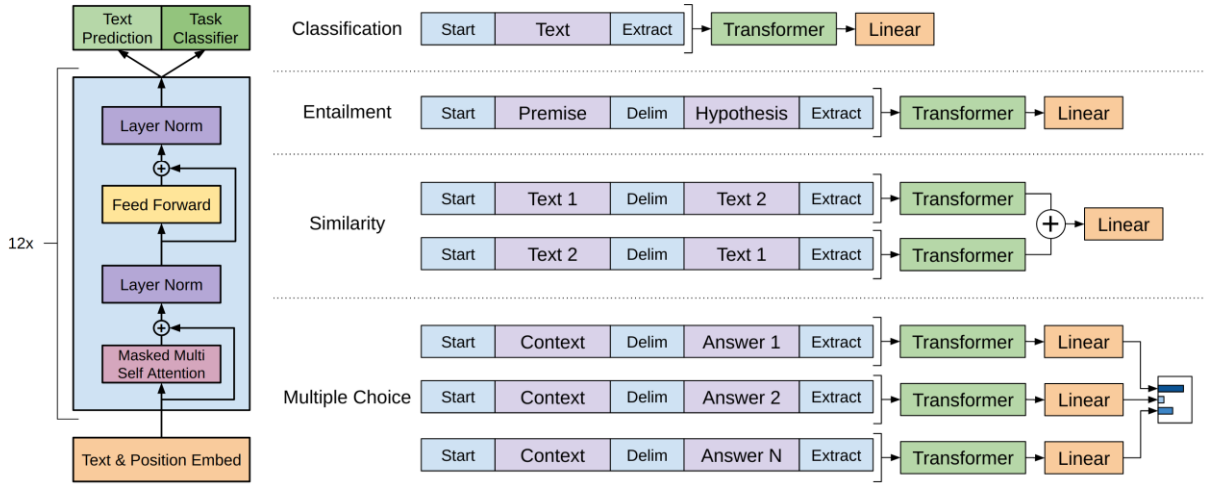

Figura 2.5: **(esquerda)** Arquitetura e objetivos de treinamentos dos modelos GPT. **(direita)** Transformações de entrada para refino do modelo em diferentes tarefas. (Radford and Narasimhan, 2018).

Conforme Radford and Narasimhan (2018), a arquitetura GPT é baseada em *Transfor-*

mers e utiliza uma abordagem semi-supervisionada durante o treinamento. Inicialmente é realizada uma aprendizagem não-supervisionada com base na tarefa de modelagem da linguagem (*language modeling*). Dessa forma, o modelo é capaz de gerar textos com certo grau de coerência. Com base no modelo de linguagem obtido, na etapa de aprendizagem supervisionada (refino), os parametros aprendidos anteriormente são utilizados com novas entradas adaptadas para o objetivo de realizar tarefas específicas com base no conceito de *transfer learning*, ou transferência de aprendizado.

Ainda em relação a estratégia de treinamento semi-supervisionado, o autor argumenta que a habilidade de aprender a partir de textos brutos é essencial para aliviar a dependência do aprendizado supervisionado no Processamento de Linguagem Natural (PLN). O mesmo argumenta ainda que o aprendizado profundo costuma depender de uma grande quantidade de dados manualmente rotulados, o que acaba restringindo a sua aplicabilidade em vários domínios que sofrem de escassez de dados anotados. Além disso, a experiência utilizando *word embeddings* (representações de palavras como vetores multidimensionais) pré-treinados indica que aprender boas representações de maneira não supervisionada melhora a perfomance dos modelos em tarefas supervisionadas.

Conforme a figura 2.5, a arquitetura final (esquerda) tem como objetivos gerar uma representação do texto de entrada (*text prediction*) e também gerar uma classificação da tarefa associada a entrada. Na parte direita da figura é mostrada a estruturação de entradas para realização de diferentes tarefas a partir do modelo pré-treinado. Para a tarefa de classificação, é necessário somente um *token* indicando o início do texto (*start*) e um *token* indicando o fim do texto (*extract*).

Para a tarefa de comparação de similaridade entre textos, são aplicadas duas entradas ao modelo compostas pelo *token* de início, os textos a serem comparados separados por um delimitador e em seguida um *token* de finalização. A ordem dos textos a serem comparados é alternada entre as entradas pois não há ordem entre os textos a ser levada em consideração durante a comparação. O modelo pré-treinado produz para cada entrada uma sequência de representações que são adicionadas elemento a elemento, formando um vetor único. Esse vetor produzido alimenta uma camada de saída linear.

2.2.10 *Bidirectional Encoder Representations from Transformers* (BERT)

Para a arquitetura BERT é proposto um modelo bidirecional de *transformer*. Desse modo, o modelo é capaz de aprender com base nas sequências ordenadas da esquerda para a direita e também de maneira inversa, tendo uma maior quantidade de informação contextual para realizar a atualização de parâmetros.

Uma diferença fundamental entre os modelos de *transformers* é que o BERT utiliza como objetivo durante a fase não supervisionada o conceito de *masked language model*

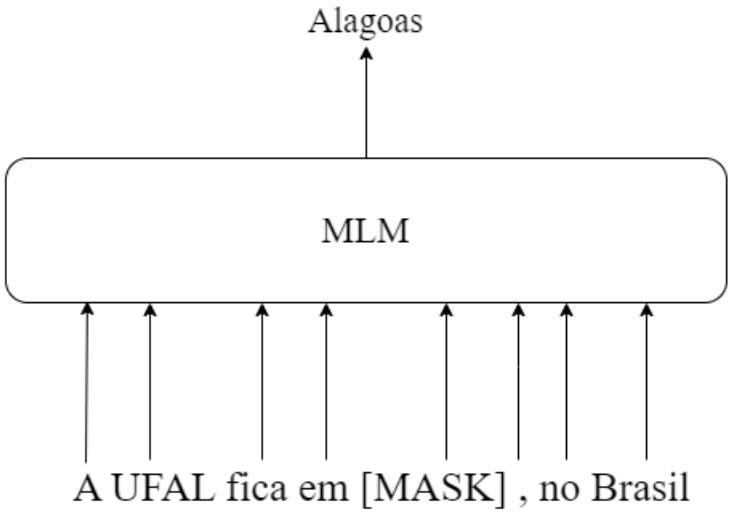

Figura 2.6: Representação de objetivo de modelo *Masked Language Model* (Fonte: autor).

(MLM). Esses tipo de modelo oculta aleatoriamente partes dos *tokens* de entrada e os seus objetivos de treinamento são de prever a palavra correta que foi ocultada baseado somente no contexto, conforme a figura 2.7 ilustra. No artigo original, 15% dos *tokens* de cada sequência de treinamento foram ocultados aleatoriamente.

Além disso, na etapa de refino (*fine-tuning*), quando treinado para tarefas como perguntas e respostas e inferência de linguagem natural, foi adotada como tarefa intermediária a predição da próxima sentença. Nessa tarefa, duas sentenças A e B são escolhidas, de modo que o objetivo é que o modelo defina se B é a sentença seguinte a A.

2.2.11 Large Language Models

Os modelos de linguagem surgiram com o objetivo da geração de textos coerentes. Posteriormente, esse tipo de modelo passou a ser desenvolvido com o objetivo de produzir modelos generalistas capazes de realizar múltiplas tarefas por meio de etapas de refino por aprendizado supervisionado. Anteriormente eram desenvolvidos modelos específicos para tarefas como tradução, sumarização, extração de informação, entre outras tarefas. Dentre as características em comum para esses modelos está também o treinamento realizado em uma quantidade de dados massiva.

De acordo com Naveed et al. (2024), os modelos LLM além de demonstrarem uma maior capacidade de generalização e adaptação do domínio, aparentam boas capacidades em tarefas como compreensão, planejamento, tomada de decisão, aprendizado dentro de contexto, dentre outras. Essas habilidades são adquiridas devido a quantidade de dados utilizados para o treinamento durante a fase não supervisionada.

Conforme OpenAI et al. (2024), os modelos de LLM baseados na arquitetura GPT não são totalmente confiáveis, pois podem sofrer de “alucinações”, possuem uma janela de contexto limitado e não aprendem a partir da experiência. Para Maynez et al. (2020), em

aprendizado profundo, o modelo "alucinar" está relacionado com a produção de conteúdo inverídico ou sem sentido de acordo com certas fontes. O autor menciona outras arquiteturas de redes neurais possuem características semelhantes.

2.2.12 Prompt engineering

Enquanto o BERT é um modelo LLM disponível para download e refino, alguns modelos de LLM são proprietários e estão disponíveis ao público somente para uso enquanto *chatbot*, como é o caso do ChatGPT. Dessa forma, esses modelos não podem ser treinados ou refinados pelo usuário, somente utiliza-se as capacidades já disponíveis do modelo. Para realização de uma tarefa específica e otimização dos resultados utilizando o ChatGPT é utilizada a técnica conhecida como "engenharia de comandos" ou *prompt engineering* para extrair as melhores informações do modelo.

Essa técnica surgiu tendo em vista que os modelos *transformers* atribuem representações vetoriais para cada *token* levando em consideração vetores de atenção desse *token* em relação aos outros na etapa de *input embedding*. Conforme Russe et al. (2024), a etapa de *embedding* da entrada desempenha um papel fundamental pela sua capacidade de capturar o contexto, a semântica e as relações com os dados originais. Com isso, o ajuste das palavras do texto de entrada e do contexto da conversa em modelos LLM permite gerar saídas mais precisas e úteis, sendo uma etapa importante para a melhora das métricas de avaliação na maioria das tarefas.

Para Russe et al. (2024), os *prompts* devem compreender as seguintes informações:

- **Tarefa e papel:** Nomear de maneira concisa a tarefa a ser desempenhada e o papel do modelo de LLM;
- **Configuração contextual:** Fornecer o contexto relevante ou informações básicas para habilitar a realização da tarefa com poucos ou nenhum dado de exemplo;
- **Instruções específicas da tarefa:** Fornecer instruções diretas e específicas a cerca da tarefa;
- **Especificações de formatação de saída**: Definir como deseja-se que a saída seja estruturada;
- **Notas de advertência:** Incluir notas de advertência para prevenir que a tarefa fique mais complicada que o necessário ou que seja mal entendida.

2.3 Reconhecimento de entidades nomeadas

A tarefa de reconhecimento de entidades nomeadas foi inicialmente proposta por Chinchor and Robinson (1998) tendo como objetivo identificar todas as instâncias de alguns

"tipos de expressão". O sistema deveria produzir uma única e inequívoca saída para cada instância com a referência de qual tipo de expressão se referia.

A definição fornecida por Borthwick et al. (1998) é identificar e categorizar todos os membros de certas categorias de nomes próprios. Alguns exemplos de entidades nomeadas são pessoas, organizações, localizações, datas, países, entre outros. Segundo Li et al. (2022), uma entidade nomeada é uma palavra ou frase que claramente identifica um item dentro de um conjunto de outros itens que possuem atributos similares.

Dentre as abordagens para realização dessa tarefa, destacam-se as abordagens baseadas em regras, abordagens baseadas em aprendizado não supervisionado e as abordagens supervisionadas.

O reconhecimento de entidades nomeadas tem servido como uma etapa de pré-processamento para uma grande variedade de tarefas, como recuperação de informação, perguntas e respostas, tradução por máquina, busca semântica, entre outras (Li et al., 2022).

2.3.1 Avaliação de modelos

Para Segura-Bedmar et al. (2013), as principais formas de avaliar modelos de reconhecimento de entidades nomeadas são:

- **Combinação estrita (*Strict*):** todos os *tokens* da entidade nomeada foram identificados corretamente sem exceder para outros tokens. Além disso, o tipo de entidade foi identificados corretamente para todos os tokens;
- **Combinação de limites exata (*Exact*):** se todos os tokens da entidade nomeada foram identificados sem exceder ainda que o tipo tenha sido identificado incorretamente;
- **Combinação de limites parcial (*Partial*):** se parte dos tokens da entidade nomeada foram identificados ainda que o tipo tenha sido identificado incorretamente;
- **Combinação de tipo (*Type*):** se pelo menos alguns tokens da entidade nomeada foram designados ao tipo correto.

Essas métricas podem ser utilizadas em conjunto com as métricas introduzidas por Chinchor and Sundheim (1993), onde a classificação de cada token fornecido pelo modelo é comparada com a classificação esperada (*golden*) individualmente. A comparação segue a seguinte lógica:

- **Correta (COR)**: mesma classificação entre modelo e esperado;
- **Incorreta (INC)**: classificação diferente entre modelo e esperado;

- **Parcial (PAR)**: parte da entidade foi classificada corretamente;
- **Ausente (MIS)**: o modelo não classificou um token quando deveria;
- **Espúria (SPU)**: o modelo classificou um token quando não deveria.

Além disso, para avaliação dos modelos, utilizamos o número de classificações de tokens esperadas e o número de classificações produzidas pelo modelo:

$$\text{ESPERADOS} = \text{COR} + \text{INC} + \text{PAR} + \text{MIS} = \text{TP} + \text{FN} \tag{2.6}$$

$$\text{PRODUZIDOS} = \text{COR} + \text{INC} + \text{PAR} + \text{SPU} = \text{TP} + \text{FP} \tag{2.7}$$

Uma análise acerca dos métodos de avaliação de modelos de extração de informação (incluindo reconhecimento de entidade nomeadas) é realizada por Esuli and Sebastiani (2010), exemplificando vantagens e desvantagens de cada forma mencionada anteriormente. Os autores argumentam que a métrica de pontuação F_1 (F_1 *score*) possui propriedades interessantes como não depender de verdadeiro negativos, que costumam conter números altos em tarefas como reconhecimento de entidades nomeadas. Ao final, os autores propoem o uso da métrica F_1 com micromédias.

2.4 Redes complexas

Conforme Albert and Barabási (2002), redes complexas são grafos que podem representar sistemas da natureza ou sociedade que possuem principios de organização robustos. Dentre os exemplos mais mencionados estão o arranjo celular, que seguem princípios complexos para gerar o corpo humano, a Internet e as redes sociais. Devido a facilidade em modelar relações por meio de grafos, surge a utilização das ferramentas fornecidas pela matemática para extrair e/ou demonstrar informações representadas em padrões que, por vezes, só podem ser observadas em análises em maior escala.

2.4.1 Grafos

Um grafo é um par de conjuntos $G = \{P, E\}$ onde P é um conjunto de N nós e E é um conjunto de arestas que conectam dois elementos de P. Um grafo G_1 contendo um conjunto de nós P_1 e arestas E_1 é um subgrafo de $G = \{P, E\}$ se todo nó em P_1 também são nós de P e toda aresta em E_1 também são arestas de E.

Centralidade de nós em grafos

A métricas de centralidade em grafos buscam prover ferramentar para identificar nós de maior influência. Dentre as medidas de centralidade em grafos, podemos destacar:

- **Centralidade de grau:** fração de nós de um grafo que estão ligados a um determinado nó ou quantidade de arestas incidentes
- **Centralidade de intermediação:** fração de caminhos mais curtos entre pares de nós que passam por um determinado nó.

De acordo com Golbeck (2015), apesar de uma maior centralidade de grau indicar um número maior de conexões do nó, a medida por si só não é capaz de mostrar qual nó é mais influente em uma rede complexa. Já a centralidade de intermediação é capaz de representar o quanto um nó atua como intermediário nas conexões entre outros nós.

2.4.2 Representações circulares em grafos

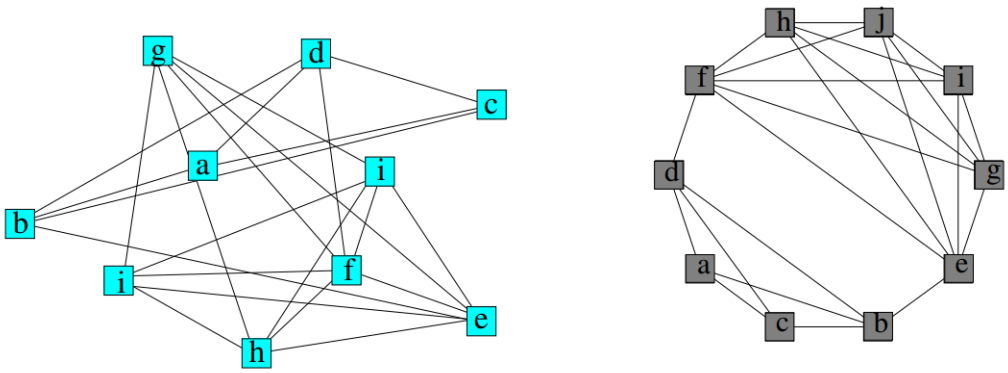

Figura 2.7: Um grafo com coordenadas arbitrárias para os nós e uma representação circular do mesmo grafo quando aplicado ao algoritmo proposto por Six and Tollis (1999) (Fonte: Six and Tollis (1999)).

Six and Tollis (1999) apresenta um algoritmo para a geração de representações circulares em grafos. Os autores argumentam que, em representações circulares, os grafos devem ser particionados em agrupamentos (ou clusters), os nós de cada agrupamento devem ser postos sob uma circunferência de *embedding* e cada aresta é desenhada como uma linha reta.

2.4.3 Representações orgânicas em grafos

Fruchterman and Reingold (1991) propõe uma representação para grafos não direcionados utilizando arestas por meio de linhas retas baseada na simulação de sistemas físicos. Dentre os objetivos estéticos para a representação, os autores destacam que a representação deve ter uma distribuição igualitária dos nós na figura, deve minimizar o cruzamento entre arestas, ter arestas de tamanhos uniformes, refletir simetrias inerentes ao grafo e se ajustar a figura.

2.4.4 Detecção de comunidades

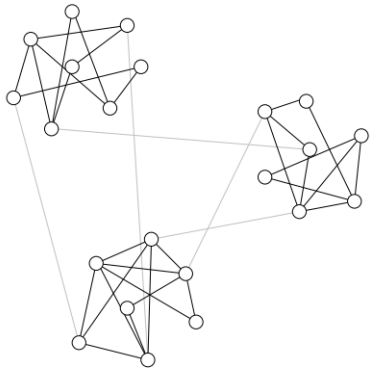

Figura 2.8: Representação de uma rede contendo estruturas de comunidade (Fonte: (Girvan and Newman, 2002)).

Girvan and Newman (2002) propõe um método para detectar o que ele chama de comunidades, ou seja, grupos de nós em uma rede complexa que são mais fortemente conectados entre si em comparação com outros nós da rede. Como forma de demonstrar o seu método, ele aplica o algoritmo a um grafo que representam as relações de amizades entre membros de um clube e um grafo representando partidas entre times de futebol americano, dentro outros grafos utilizados. Ambos os grafos mencionados anteriormente foram escolhidos por possuirem estruturas previamente conhecidas para que seja possível a validação do método enquanto forma de extração de informação de redes complexas.

Dentre as principais aplicações, o autor destaca a análise de colaborações entre grupos de pesquisadores de diferentes áreas, o que pode ter enquanto resultado um incentivo a colaboração entre outros pesquisadores que ainda não trabalham juntos, e a análise de grupos ecológicos.

O autor apresenta a modularidade como métrica para estimar a qualidade da divisão de comunidades em um grafo. A métrica é uma medida da diferença de densidade de conexões entre os nós da comunidade e a densidade média de uma rede aleatória.

Capítulo 3

Trabalhos relacionados

3.1 Reconhecimento de entidades nomeadas

Korkontzelos et al. (2015) utiliza um dicionário de conhecimento incluindo regras para realizar o reconhecimento do nome de fármacos sem a necessidade de uma grande quantidade de anotações manuais. Além disso, uma lista de fármacos conhecidos é utilizada para melhorar a perfomance. Desse modo, os autores conseguem atingir um F-score de 95%.

Zhou and Su (2002) utiliza um modelo Hidden Markov Model (HMM) para identificar entidades nomeadas. Patil et al. (2020) utiliza o modelo estatístico *Conditional Random Fields* (CRF) para reconhecer entidades nomeadas utilizando aprendizagem supervisionada. O modelo CRF posteriormente foi utilizado em conjunto com arquiteturas de redes neurais recorrentes para a realização dessa mesma tarefa.

Liu and Zhou (2013) utiliza uma abordagem baseada em duas etapas. A primeira etapa se baseia em aprendizado supervisionado, utilizando o modelo *Conditional Random Fields* (CRF), onde é obtido um modelo rotulador de sequências para identificar localizações, países, expressões de tempo, entre outras entidades. A segunda etapa funciona por aprendizado não-supervisionado. Nessa etapa as saídas do modelo e as rotulações manuais passam por um algoritmo de agrupamento de modo que textos com palavras parecidas ficam próximas. Assim, para cada agrupamento, as rotulações fornecidas pelo modelo supervisionado são ajustadas para ficar de acordo com as demais rotulações presentes no agrupamento.

Singh et al. (2009) propõe a utilização de Support-Vector Machines (SVM) para identificação de localizações, pessoas e organizações na língua Manipuri. Na literatura é possível observar um grande destaque para arquiteturas de redes neurais recorrentes *Long-Short Term Memory* (LSTM), em especial as arquiteturas bidirecionais, como a arquitetura ELMo (Peters et al., 2018).

Cho et al. (2020) propõe uma arquitetura convolucional em conjunto com uma rede

neural recorrente bidirecional e um modelo CRF (CNN-BiLSTM-CRF) que utiliza representações vetoriais em nível de caractere. As representações vetoriais em nível de caractere buscam trazer ao modelo uma maior robustez ao modelo contra palavras que não apareceram durante o treinamento.

Arquiteturas baseadas em *Transformers*, como BERT (Devlin et al., 2019), conseguiram superar as redes neurais recorrentes em comperações para diversos conjuntos de dados, se estabelecendo enquanto estado da arte desde então. Uma combinação entre o modelo BERT e redes neurais LSTM (ELMo) é demonstrada por Affi and Latiri (2021).

Wang et al. (2023) apresenta uma abordagem para transformar a tarefa de reconhecimento de entidades nomeadas em uma tarefa de geração de textos que pode ser utilizada em modelos LLM, como o método proposto no presente trabalho. O modelo escolhido para realização dos experimentos é o GPT-3. O autor realiza uma comparação entre os resultados da abordagem sem fornecimento de exemplos ao modelo (*zero shot*) e a abordagem fornecendo 3 exemplos. A seleção dos 3 exemplos é feita por meio de uma estratégia de buscar textos semelhantes a partir de representações vetoriais do texto e do algoritmo k-NN. Para lidar com o problema de alucinação, que é observado em modelos LLM, o autor propôe uma auto-verificação, onde é solicitada uma confirmação ao modelo se cada entidade nomeada apontada pertence de fato ao tipo de entidade predito.

3.2 Geração de redes complexas utilizando Reconhecimento de Entidades Nomeadas

O trabalho proposto por Sawicki et al. (2024) busca reconhecer entidades nomeadas na rede social *Reddit* e as uitiliza para formar um grafo de relacionamento entre comunidades dentro da rede social (*"subreddits"*). O autor utiliza um modelo BERT refinado para o reconhecimento de entidades específicas das comunidades trabalhadas no artigo.

Fan et al. (2020) buscar criar um grafo de conhecimento sobre perigos geológicos para facilitar o reuso da literatura acerca do tema e prover uma referência de governança. O artigo utiliza uma arquitetura de rede neural recorrente bidirecional combinada com uma camada de *Conditional Random Fields* (BiGRU-CRF). As entidades trabalhadas no artigo são localizações, métodos e dados. O grafo foi construído de modo em que as entidades nomeadas são vertices e quando aparecem em um mesmo artigo são ligadas por meio de uma aresta.

O artigo desenvolvido por Wei et al. (2023) propôe a utilização do reconhecimento de entidades nomeadas em artigos publicados para construir um grafo de conhecimento sobre a técnica de perfuração por pressão controlada (MPD). A técnica referida é utilizada na indústria de óleo e gás. Os autores propoem um modelo de aprendizado com poucos exemplos (*few-shot learning*) para a extração de entidades nomeadas que é combinado

com a extração de relações por meio de regras pré-definidas. Ao total, o modelo busca reconhecer 14 entidades nomeadas que correspondem a etapas do processo de perfuração. Para o grafo de conhecimento obtido, os vértices são processos realizados (identificados por serem entidades nomeadas) ligados por meio de arestas a outros processos quando reconhecidos no texto, nomeando a aresta com a informação de qual etapa o processo se refere.

3.3 Detecção de comunidades

Ribeiro et al. (2018) analisa escândalos de corrupção na política brasileira por meio de redes onde dois indivíduos são conectados por arestas se foram envolvidos em um mesmo escândalo. Os autores argumentam que a estrutura da rede de corrupção política pode ser utilizada para prever com sucesso parceiros em escândalos futuros. Os autores foram capazes de encontrar 27 comunidades (sendo 13 comunidades desconexas em relação ao grafo principal) dentro do grafo contendo 404 nós e 3549 arestas. Também foi possível demonstrar o efeito mundo pequeno, onde a distância entre duas pessoas mencionadas em escândalos era de, em média, $2,99$ passos.

3.4 Contribuições em relação aos trabalhos existentes

Dentre as contribuições do presente trabalho vale destacar que o método proposto por Wang et al. (2023) realiza o reconhecimento de um tipo de entidades nomeada por entrada, enquanto o método proposto no presente trabalho é capaz de identificar diferentes tipos de entidades nomeadas com apenas uma entrada, reduzindo significativamente a quantidade de tokens fornecidos ao modelos para o desempenho da tarefa. Além disso o método busca aproveitar a janela de contexto, disponível para o serviço ChatGPT, para reduzir a quantidade de tokens utilizados.

Capítulo 4

Metodologia

Nesta seção será descrito o método utilizado para realização do reconhecimento de entidades nomeadas, obtenção do grafo de relações entre entidades nomeadas e para realização das análises dos grafos obtidos. A seção 4.1 descreve a etapa de reconhecimento de entidades nomeadas utilizando arquiteturas de redes neurais BiLSTM e BERT, enquanto a seção 4.2 descreve mais especificamente o reconhecimento de entidades nomeadas utilizando *Large Language Models* e engenharia de *prompt*, por meio do modelo ChatGPT 3.5. A divisão das seções é feita devido a disponibilidade de código e pesos treinados para a BiLSTM e o BERT, enquanto para o ChatGPT a tarefa é realizada por meio de *prompt engineering*.

A seção 5.1, dedicada a apresentar o conjunto de dados utilizado nos experimentos, detalha a etapa de pré-processamento dos dados textuais necessária para a aplicação do método.

4.1 Reconhecimento de entidades nomeadas com redes neurais

Para realização da análise quantitativa dos modelos de LLMs na realização da tarefa de reconhecimento de entidades nomeadas foram comparados os resultados de 3 modelos: redes neurais BiLSTM, BERT e ChatGPT 3.5.

O modelo BERT foi escolhido por ter sido base para modelos que constituem o estado da arte na tarefa de reconhecimento de entidades nomeadas, como as arquiteturas RoBERTa (Conneau et al., 2020) e DistilBERT (Sanh et al., 2020).

O modelo de rede neural BiLSTM é utilizado somente enquanto *baseline* para análise quantitativa dos resultados fornecidos pelo BERT e pelo ChatGPT, mas não é gerado grafo de relações entre entidades nomeadas a partir dessa arquitetura. O objetivo da utilização da arquitetura BiLSTM enquanto modelo *baseline* está na comparação de modelos pré-treinados em grandes conjuntos de dados (BERT e ChatGPT) com uma arquitetura

tradicional treinada somente com os exemplos disponíveis no conjunto de dados rotulado manualmente.

O treinamento do modelo BiLSTM foi feito utilizando a biblioteca de computação matemática Tensorflow[1] (versão 2.15.0). Já para o refino do modelo BERT foi utilizada a biblioteca Transformers[2] (versão 4.42.4) disponibilizada pelo site Hugging Face[3]. O Hugging Face armazena pesos e conjuntos de dados (públicos ou não) para treinamento de uma grande variedade de arquiteturas de redes neurais, em especial, arquiteturas *transformer*. Além disso, o site fornece implementações em Python para a realização de diversas tarefas de aprendizado de máquina. No site estão disponibilizados abertamente os pesos do BERT[4] obtidos a partir do treinamento em grandes conjuntos de dados realizado pela Google.

4.2 Reconhecimento de entidades nomeadas utilizando engenharia de *prompt*

A principal vantagem do uso do ChatGPT para as tarefas de processamento de linguagem natural está na capacidade de realização utilizando poucos (ou até nenhum) exemplos de entrada e saída. Para desempenhar o presente trabalho foi fornecido, além das instruções, somente um exemplo de entrada e saída esperada. Essa característica de realizar a tarefa com poucos exemplos é de grande valia quando novos dados surgem no dia a dia. Modelos BiLSTM e BERT necessitam de um treinamento ou refino contendo uma quantidade considerável de exemplos e recursos. Ao surgirem dados com situações não encontradas anteriormente, esses modelos necessitam de um novo treinamento para que possam desempenhar a tarefa. A obtenção de exemplos em grande quantidade é um desafio especialmente em temas mais especializados.

Para avaliação quantitativa das respostas fornecidas pelo ChatGPT, foram utilizadas todas as rotulações manuais utilizadas para o treinamento dos outros modelos.

Levando em consideração as sugestões para construção de *prompts* apresentadas por Russe et al. (2024), dentre as etapas para realizar o reconhecimento, primeiramente foi feita a iniciação do contexto da conversa por meio do seguinte *prompt*:

```
--
Instruction: You're a Named Entity Recognition Specialist. Perform a
Named Entity Recognition task. These are the labels entities to
be identified:
- COUNTRY for country names or abbreviations entities. Example: United
```

[1] https://www.tensorflow.org/
[2] https://huggingface.co/docs/transformers/index
[3] https://huggingface.co/
[4] https://huggingface.co/docs/transformers/model_doc/bert

```
States, USA, US, Russia.
- ORG_GOV for governmental organizations. Example: Senate, FBI, European
Union, eu.
- ORG_N_GOV for non-governmental organizations. Example: Google, Meta,
OpenAI.
- LAW for laws and treaties. Example: Article 5, Constitution.
- LOC for locations that are not country names. Example: South Asia, Europe.
- PERSON for person name. Example: Barack Obama.
- ROLE for job positions. Example: Journalist, Representative,
Senator, President.

Output only the answer. Print only entities found. When there is no
entities, output "~ No entities found ~"

Consider the following formatting example:

Input:
The Indian President Doupadi Murmu made a deal with Hyundai to open
a new car factory in Mumbai.
Output:
COUNTRY: Indian # ROLE: President # PERSON: Doupadi Murmu #
ORG_N_GOV: Hyundai # LOC: Mumbai
--
```

O objetivo desse *prompt* é contextualizar qual a tarefa a ser realizada, dar uma breve explicação dos critérios para as entidades nomeadas que forem ambíguas e então gerar um exemplo da formatação de saída esperada para viabilizar o processamento posterior.

Em seguida, cada texto ao qual deseja-se reconhecer as entidades nomeadas é fornecido por meio do seguinte *prompt*:

```
--
Return all COUNTRY, ORG_GOV, ORG_N_GOV, LAW, LOC, PERSON and ROLE
entities. Here is an example:
Input:
The Indian President Doupadi Murmu made a deal with Hyundai to open a new
car factory in Mumbai.
Output:
COUNTRY: Indian # ROLE: President # PERSON: Doupadi Murmu # ORG_N_GOV:
Hyundai # LOC: Mumbai
```

```
Input: {texto a ser reconhecido}
--
```

Nesse *prompt* é mencionado novamente as entidades para que a atenção volte a considerar o contexto anterior sempre que uma nova entrada for fornecida. Note que o primeiro *prompt* só é utilizado para iniciação da conversa. As entradas posteriores fazem uso somente do mecanismo de atenção e do contexto. Ao reiterar os tipos de entidades nomeadas que o modelo deve reconhecer no segundo *prompt*, buscou-se evitar uma alucinação apresentada pelo modelo onde uma entidade não especificada, denominada "EVENT", era utilizada para reconhecer trechos da entrada que se referiam a eventos históricos, como a 2ª Guerra Mundial.

O modelo produz como saída somente as entidades nomeadas que foram identificadas e suas respectivas classificações separadas por #, o que já é suficiente para a construção da rede de relações entre entidades. A formatação de saída utilizada visou simplificar a resposta esperada a fim de evitar situações em que a resposta é produzida em formatos que dificultam o processamento automático.

As respostas produzidas pelo ChatGPT passaram posteriormente por uma etapa de conversão em sequências de rótulos IOB para permitir a avaliação quantitativa em comparação com com os modelos BiLSTM e BERT. A conversão foi realizada por meio de uma busca da entidade nomeada reconhecida pelo modelo no texto original. A conversão das respostas do ChatGPT em sequências de rótulos IOB permitiu também a similaridade entre as etapas posteriores ao reconhecimento de entidades nomeadas.

O ChatGPT possui um serviço pago de API com biblioteca facilitadora disponível em Python. No entanto, devido aos limites de uso da API[5] no período de realização dos experimentos a abordagem utilizada foi a construção de um script Python em conjunto com um software de automação capaz de fornecer entradas ao modelo e copiar as saídas.

Dentre as particularidades do uso do ChatGPT, destaca-se a produção de algumas saídas contendo correções ortográficas das entidades nomeadas identificadas. Para permitir que o método se adapte a essa característica e possibilitar a conversão das saídas produzidas pelo modelo em sequências de rótulos IOB, foi utilizada a distância de Levenshtein (descrita na seção 2.1.1) para tornar o método robusto a pequenas diferenças ortográficas, como exemplificado a seguir:

[5] `https://platform.openai.com/docs/guides/rate-limits/usage-tiers`

ChatGPT	Texto original
Commander-in-Chief	CommanderinChief
Lokpal Act	Lokpal
Ukrainian	Ukranian
Indian-Americans	IndianAmericans

Tabela 4.1: Exemplos de equivalências identificadas por meio da distância de Levenshtein.

O objetivo de separar os prompts é evitar o limite de *tokens* do serviço, que é estabelecido por mensagem, por hora e por dia, porém fornecendo ao contexto informações suficientes para estabelecer um critério comum entre a rotulação manual e o esperado pelo modelo.

4.3 Elaboração de rede de relações entre entidades nomeadas

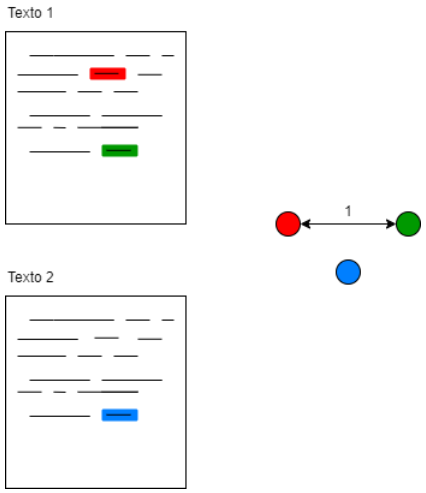

Figura 4.1: **(esquerda)** Textos contendo entidades nomeadas representadas em vermelho, verde e azul. **(direita)** Grafo de relações entre entidades nomeadas obtido apartir da aplicação do algoritmo 1 aos textos 1 e 2 (Fonte: autor).

Após o modelo de reconhecimento de entidades ser treinado, foi utilizado para montar uma rede com relações entre as entidades nomeadas. A rede é modelada por meio de um grafo não direcionado onde os nós representam entidades nomeadas e as arestas representam ocorrências de entidades nomeadas em um mesmo texto. Para cada entidade presente em um mesmo texto, é adicionada uma aresta de peso 1 com as outras entidades nomeadas do texto. Sempre que as entidades nomeadas voltam a aparecer juntas em outros textos, é adicionado 1 ao peso da aresta entre os nós das entidades. A montagem da rede de relações é melhor detalhada por meio do pseudo-código 1.

A figura 4.1 representa a aplicação do algoritmo de elaboração da rede de relações entre entidades nomeadas quando aplicados a dois textos preditos pelos modelos de NER. As palavras pertencentes a entidades nomeadas estão representadas nas cores vermelho, azul e verde. Cada entidade nomeada identificada gera um nó no grafo e ocorrências em um mesmo texto geram arestas onde o peso representa a quantidade de textos da relação.

Algorithm 1 Montagem de grafo de relações entre entidades nomeadas

$G \leftarrow$ Novo Grafo
$similaridades\_conhecidas \leftarrow$ Dicionário de similaridades entre palavras
$palavras\_ignoradas \leftarrow$ Lista palavras a serem ignoradas
$textos\_pred \leftarrow model.predict(textos)$
for $texto, labels \in textos\_pred$ **do**
    $lista\_entidades\_nomeadas \leftarrow separaEntidades(texto, labels)$
    $lista\_entidades \leftarrow$ Lista vazia
    for $entidade \in lista\_entidades\_nomeadas$ **do**
        $texto\_entidade, label \leftarrow entidade$
        if $texto\_entidade$ em $palavras\_ignoradas$ **then**
            $Pula$
        end if
        if $texto\_entidade \in list(similaridades\_conhecidas.chaves())$ **then**
            $texto\_entidade \leftarrow similaridades\_conhecidas[texto\_entidade]$
        end if
        $lista\_entidades.add((texto\_entidade, label))$
    end for
    for $entidade \in lista\_entidades$ **do**
        if $entidade \notin G$ **then**
            $G.adiciona\_no(entidade, valor = 1)$
        else
            $G.no(entidade)[\text{“}valor\text{”}] += 1$
        end if
    end for
    for $no_1, no_2 \in list(combinations(lista\_entidades, 2))$ **do**
        if Não existe aresta entre no_1 e no_2 **then**
            $G.adiciona\_aresta(no_1, no_2, peso = 1)$
        else
            $G.aresta(no_1, no_2)[\text{“}peso\text{”}] += 1$
        end if
    end for
end for

A biblioteca utilizada para criação e manipulação da rede de relações é a Networkx (Hagberg et al., 2008), disponível em Python. Essa biblioteca provê funções que facilitam a exploração, a análise e a visualização de redes. Para a visualização de grafos foi utilizado o software de código aberto Cytoscape [6]. O Cytoscape permite gerar de maneira facilitada uma grande diversidade de layouts em grafos, como os layouts circulares e orgânicos, que foram utilizados nas visualizações apresentadas no presente trabalho.

4.3.1 Centralidade de nós

A análise de centralidade em grafos permite a identificação de nós com maior influência em toda a rede e nós que atuam como conexão entre diferente grupos. Além disso, pode ajudar na detecção de anomalias, principalmente quando utilizadas centralidades que atribuem maior peso ao subgrafo de conexões mais próximas. O presente trabalho considerou a análise de centralidade de grau e de intermediação.

4.3.2 Subgrafo de relações mais relevantes e subgrafo de conexões diretas de entidade nomeada

Esse subgrafo é formado por meio da obtenção de um subgrafo induzido centrado na entidade nomeada de maior frequência. O ajuste do raio do subgrafo induzido permite a visualização de relações com uma maior distância ao nó central. Dessa forma, caso o raio especificado seja 1, iremos obter os nós vizinhos da entidade que mais vezes apareceu no conjunto. Caso o raio especificado seja 2, obteremos os vizinhos da entidade e os vizinhos dos vizinhos, assim por diante.

O próximo passo para obtenção do subgrafo desejado é remover arestas com um peso menor que um valor especificado. Dessa forma, alguns nós passam a não possuirem ligações. Nós que após a remoção de arestas não possuir ligações podem ser removidos.

O subgrafo de conexões diretas é uma implementação do pseudo-código 2 onde o raio r utilizado é 1 e o nó central é escolhido conforme especificação. Enquanto o subgrafo anterior é capaz de mostrar relações com maior distância, esse subgrafo representa somente as ligações mais diretas. Desse modo, obtém-se subgrafo induzido contendo os vizinhos do nó n que apareceram juntos em um número maior de textos diferentes. O ajuste do peso mínimo para a aresta permite a seleção de conexões de maior frequência.

4.3.3 Detecção de comunidades

A etapa de detecção de comunidades é uma etapa que busca a identificação de padrões no grafo, permitindo a identificação de grupos de nós com uma maior interação entre si

[6] `https://cytoscape.org/`

Algorithm 2 Subgrafo de relações mais relevantes de uma entidade nomeada

Require: G grafo de relações entre entidades
Require: n nó da entidade nomeada com maior valor
Require: $r \geq 0$ ▷ Raio do subgrafo induzido
Require: $min\_weight \geq 0$ ▷ Peso mínimo da aresta
$T \leftarrow networkx.ego\_graph(G, n, radius = r)$ ▷ Função que obtém subgrafo induzido
$remove\_edges\_list \leftarrow []$
for $edge \in T.edges()$ **do**
    if $T[edge[0]][edge[1]][\text{“}width\text{”}] < min\_width$ **then**
        $remove\_edges\_list.append(edge)$
    end if
end for
$T.remove\_edges(remove\_edges\_list)$
$remove\_nodes\_list \leftarrow []$
for $no, grau \in dict(T.degree()).items()$ **do**
    if $grau = 0$ **then**
        $remove\_nodes\_list.append(no)$
    end if
end for
$T.remove\_nodes(remove\_nodes\_list)$

do que com o resto do grafo. A implementação foi por meio do algoritmo de Louvain[7].

Devido a grande quantidade de nós e arestas nos grafos obtidos, a análise qualitativa do algoritmo foi realizada por meio da seleção de um subgrafo obtido a partir da remoção de arestas entre entidades nomeadas que ocorreram simultaneamente em uma quantidade menor que x documentos, onde esse valor x é ajustado conforme necessário entre os grafos do BERT e do ChatGPT. Além disso, nós isolados foram removidos do grafo.

[7] `https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.louvain.louvain_communities.html`

Capítulo 5

Experimentos

Nesta seção iremos descrever o conjunto de dados utilizado para avaliação da metodologia abordada na seção 4 e como foi realizado o treinamento dos modelos de rede neural. Em seguida, a seção 5.2 realiza uma análise quantitativa dos modelos de reconhecimento de entidades nomeadas com base nas métricas descritas na seção 2.3.1. Posteriormente, uma análise qualitativa dos grafos de relações entre entidades nomeadas é exposta na seção 5.3.

5.1 Conjunto de dados

O conjunto de dados utilizado nos experimentos é composto por perguntas e respostas retiradas da comunidade de política da rede social Stack Exchange (Exchange (2024)). Essa rede social possui uma ampla variedade de tópicos em discussão, desde filosofia, ciências até dicas de culinária. Cada tópico é separado por um endereço único conhecido como comunidade. Uma característica valiosa dessa rede é o sistema de reputação de usuário e pontuação por interações. Ainda que haja a presença de moderadores, pelo grande volume de conteúdo a filtragem de interações indevidas nem sempre ocorre de maneira ágil, sendo o sistema de pontuações um modo de avaliar a qualidade da interação.

A comunidade de política foi escolhida pela sua interdisciplinaridade, tratando de temas como economia, direito, filosofia e história. Outro fator é a grande variedade de pontos de vistas das perguntas e respostas, sendo um tópico mais subjetivo e, portanto, refletindo de maneira mais pessoal a opinião dos usuários da rede.

Todo o conteúdo da rede social é disponibilizado de maneira aberta e anonimizada no website Archive.org[1] sob licença Creative Commons By SA 4.0 (Meteo Service, 2022), que permite redistribuição e adaptação para qualquer propósito, desde que contenha atribuição a Stack Exchange. Os dados incluem as publicações, informações de reputação do usuário que realizou a interação, comentários, votos, entre outras informações.

[1] `https://archive.org/details/stackexchange`

Os dados estão disponibilizados no website em formato ZIP separados por comunidade. Dentre os arquivos presentes no ZIP estão as perguntas e respostas da comunidade estruturadas em formato XML. Após o download do conteúdo da comunidade de política da rede social foi necessária uma etapa de pré-processamento realizado utilizando a biblioteca Pandas[2] da linguagem Python[3]. Por meio dessa biblioteca foi possível converter o arquivo XML em um DataFrame. Dessa forma foi possível selecionar as postagens que possuiam uma reputação positiva. Após essa etapa, foi realizado um tratamento para remoção de tags HTML e quebras de linha.

Em seguida, foram selecionadas apenas as postagens que possuiam no máximo 250 palavras. Essa limitação ocorreu devido ao limite de *tokens* presente no modelo ChatGPT durante o período de realização dos experimentos. Vale salientar que para os modelos BiLSTM e BERT, apesar de não haver a limitação do tamanho do texto, o consumo de memória durante o treinamento aumenta significativamente.

Foram utilizadas 814 postagens selecionadas aleatoriamente para a etapa de rotulação manual, além de 15416 postagens aplicadas ao modelo BERT refinado para obtenção do grafo de relações entre entidades nomeadas.

Para o reconhecimento de entidades nomeadas foram selecionadas 7 tipos de entidades a serem identificados, conforme a tabela 5.1.

Entidades Nomeadas	Descrição
COUNTRY	Nome de países
ROLE	Cargos ou funções dentro do governo, corporação ou grupo
ORG_GOV	Órgãos governamentais, partidos políticos ou empresas públicas
LAW	Leis, tratados e jurisprudências
LOC	Localizações geográficas
PERSON	Nome de pessoas
ORG_N_GOV	Nome de organizações não governamentais ou empresas

Tabela 5.1: Descrição dos rótulos escolhidos.

A partir do conjunto de dados rotulados, foram separados, dentre 814 exemplos, 610 para treino e 204 para teste dos modelos BiLSTM e BERT. Para o ChatGPT, foi fornecido como prompt apenas 1 exemplo e foram obtidas as saídas rotuladas 814 perguntas ou respostas, como será detalhado posteriormente.

[2] `https://pandas.pydata.org/pandas-docs/version/2.2.0/`

[3] `https://docs.python.org/release/3.12.2/`

A	O
República	B-COUNTRY
Federativa	I-COUNTRY
do	I-COUNTRY
Brasil	I-COUNTRY
está	O
na	O
América	B-LOC

Tabela 5.2: Exemplo de rotulos IOB.

Os dados rotulados seguiram o formato apresentado por Sang and Meulder (2003), conhecido como rótulos IOB. Nesse tipo de rotulação os textos são separados em *tokens* e cada *token* recebe um rótulo indicando se pertence ou não a uma entidade nomeada. *Tokens* rotulados como "O" não pertencem a nenhuma entidade. Rótulos contendo o prefixo "B-" são utilizados no primeiro *token* de uma entidade nomeada enquanto rótulos com o prefixo "I-" são utilizados para os *tokens* seguintes de uma mesma entidade nomeada, caso houver.

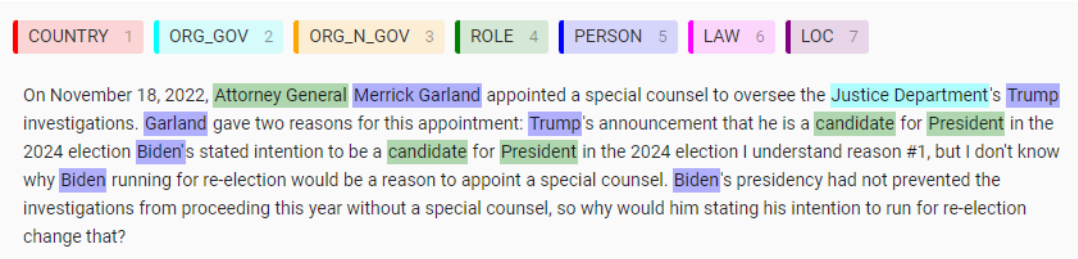

Figura 5.1: Interface gráfica do software de rotulação Label Studio (Fonte: autor).

As rotulações foram realizadas com auxílio do software de código aberto Label Studio[4] (versão 1.11.0) (Tkachenko et al. (2022)). Esse software facilita a geração de conjuntos de dados para diferentes tarefas de visão computacional e processamento de linguagem natural. A tabela 5.3 contém a quantidade de *tokens* por rótulo no conjunto de dados final.

[4] `https://labelstud.io/`

Rótulo	Treino	Teste	**Total**
O	28915	9978	38893
B-COUNTRY	721	260	981
I-COUNTRY	121	59	180
B-ROLE	361	116	477
I-ROLE	96	29	125
B-ORG_GOV	445	158	603
I-ORG_GOV	233	58	291
B-LAW	298	89	387
I-LAW	302	80	382
B-LOC	117	40	157
I-LOC	37	8	45
B-PERSON	221	46	267
I-PERSON	69	23	92
B-ORG_N_GOV	98	26	124
I-ORG_N_GOV	36	16	52
Total	32070	10986	43056

Tabela 5.3: Quantidade de *tokens* por rótulo no conjunto de dados.

5.2 Análise quantitativa

A discussão apresentada nesta seção visa mensurar a qualidade dos modelos de reconhecimento de entidades nomeadas discutidos anteriormente por meio das métricas estatísticas descritas na seção 2.3.1. A interpretação dos resultados busca estimar os pontos fortes de cada um dos modelos utilizados considerando suas particularidades de treinamento e limitações. A tabela 5.4 traz as métricas de precisão, cobertura e F_1 *score* obtidas na realização da tarefa de NER por meio dos modelos ChatGPT 3.5, BERT e BiLSTM. As figuras 5.2, 5.3, 5.4 e 5.5 apresentam visualizações das métricas de avaliação de *tokens* por combinação de tipo, combinação parcial, combinação estrita e combinação exata, respectivamente.

Métrica	**Combinações**	**ChatGPT**	**BERT**	**BiLSTM**
Precisão	Tipo	70.74%	80.07%	52.22%
	Parciais	83.87%	80.68%	49.51%
	Estrita	65.44%	74.65%	43.44%
	Exata	78.73%	77.10%	44.50%
Cobertura	Tipo	31.91%	89.98%	80.14%
	Parciais	37.84%	90.67%	75.99%
	Estrita	29.52%	83.89%	66.67%
	Exata	35.52%	86.64%	68.30%
F_1	Tipo	43.98%	84.74%	63.23%
	Parciais	52.15%	85.38%	59.96%
	Estrita	40.69%	79.00%	52.60%
	Exata	48.95%	81.59%	53.89%

Tabela 5.4: Comparação das métricas de precisão, cobertura e F_1 dos modelos ChatGPT 3.5, BERT e BiLSTM.

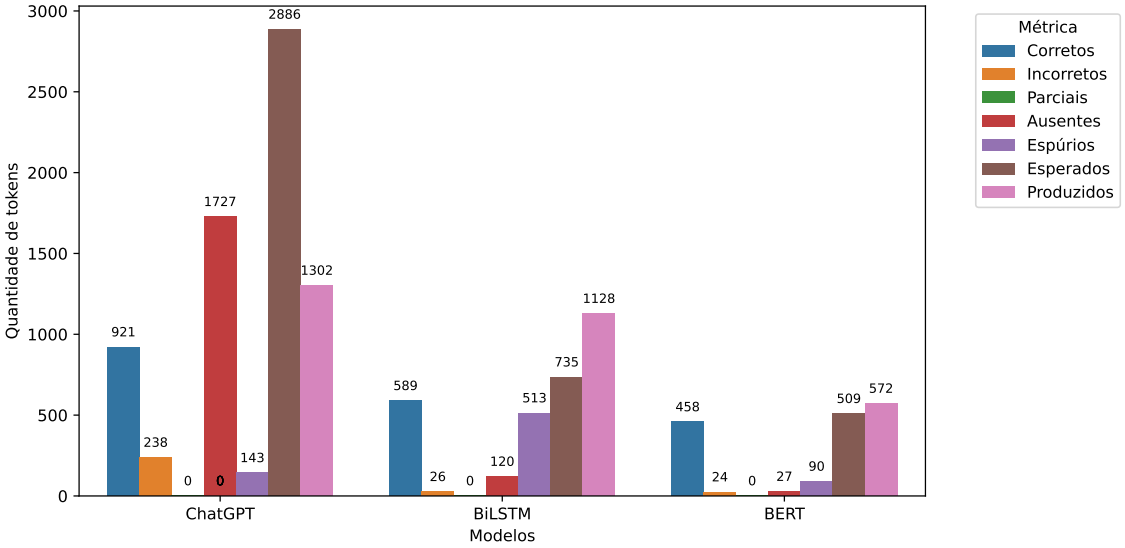

Figura 5.2: Gráfico de barras das métricas de análise de *tokens* por combinação de tipo para os modelos ChatGPT 3.5, BiLSTM e BERT (Fonte: autor).

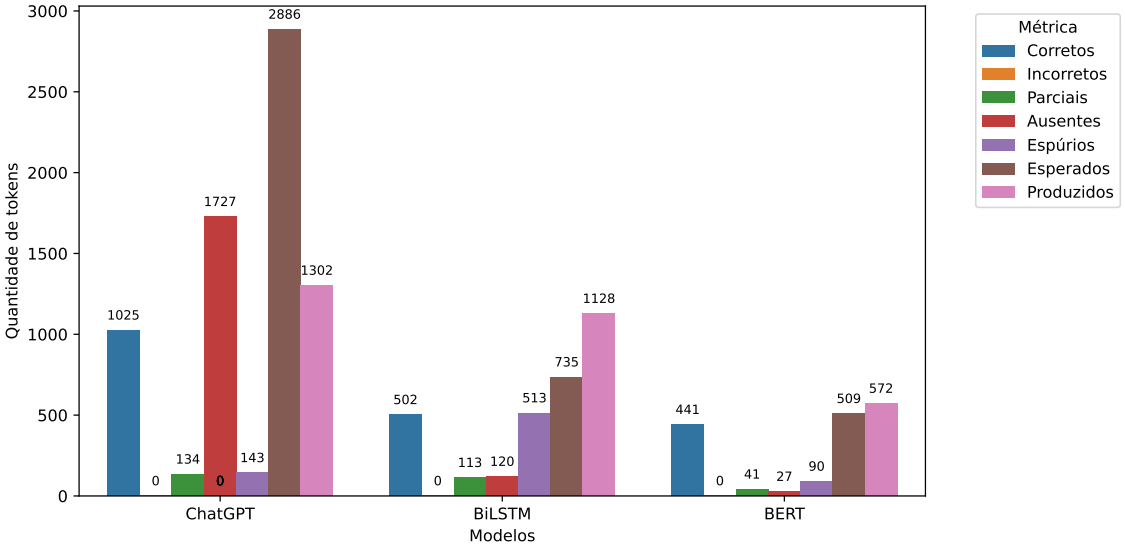

Figura 5.3: Gráfico de barras das métricas de análise de *tokens* por combinação parcial para os modelos ChatGPT 3.5, BiLSTM e BERT (Fonte: autor).

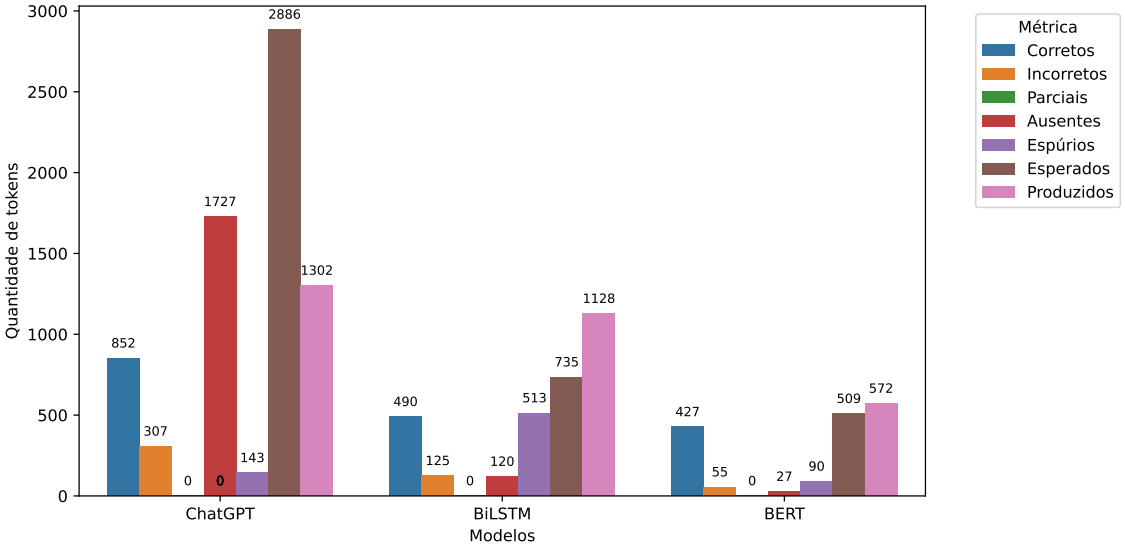

Figura 5.4: Gráfico de barras das métricas de análise de *tokens* por combinação estrita para os modelos ChatGPT 3.5, BiLSTM e BERT (Fonte: autor).

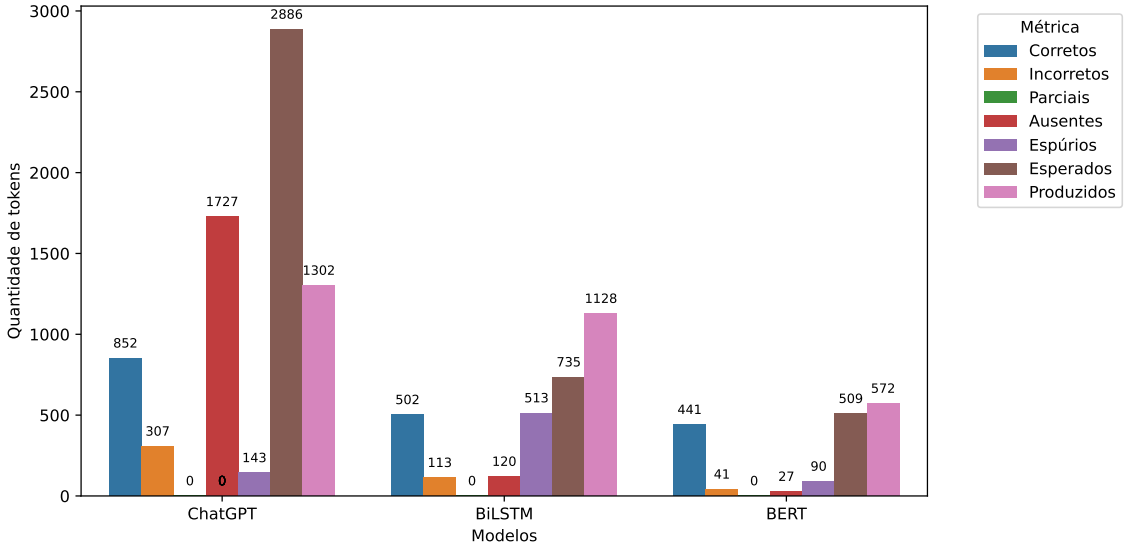

Figura 5.5: Gráfico de barras das métricas de análise de *tokens* por combinação exata para os modelos ChatGPT 3.5, BiLSTM e BERT (Fonte: autor).

Dentre os resultados observados, vale destacar que foram produzidos 393 (53.47%) *tokens* classificados além do esperado pelo modelo BiLSTM, enquanto o BERT produziu 63 *tokens* classificados além do esperado (12.37%). Já para o ChatGPT 3.5, deixaram de serem produzidos 1584 (54,88%) dos *tokens* classificados.

As diferenças entre quantidade de *tokens* esperados e produzidos refletem em uma precisão mais baixa para o modelo BiLSTM, indicando que o modelo produz uma quantidade maior de falsos positivos em relação ao BERT e o ChatGPT. Já para o ChatGPT, ocorre uma cobertura mais baixa pela ocorrência de um número maior de falsos negativos.

Como exemplo, vale mencionar que uma taxa alta de falsos negativos, como ocorre ao ChatGPT 3.5, em um contexto de moderação automática de conteúdo (em redes sociais por exemplo) pode resultar na queda de confiabilidade no serviço devido a não remoção de publicações com conteúdo inadequado, podendo resultar em uma possível perda de adesão por parte dos usuários. Além disso, a característica de gerar um número maior de falsos negativos pode ser indesejável para uma revisão automática de documentos por meio do reconhecimento de entidades.

Já o modelo BiLSTM, por gerar um número maior de falsos positivos, poderia inflar métricas de engajamento em um sistema que detecta menções a pessoas baseadas em contexto utilizando reconhecimento de entidades nomeadas.

Vale salientar que dentre as possíveis razões para a cobertura menor do ChatGPT podem estar as particularidades da rotulação manual. Apesar da breve introdução dos critérios para as entidades nomeadas durante o início do contexto, apresentado na seção 4.2, as informações ali presentes podem não ser suficientes para que o modelo de LLM consiga estabelecer um critério semelhante ao utilizado durante a rotulação manual. Essa situação poderia ser amenizada com o fornecimento de mais exemplos ao modelo ou novos

ajustes ao *prompt* fornecido.

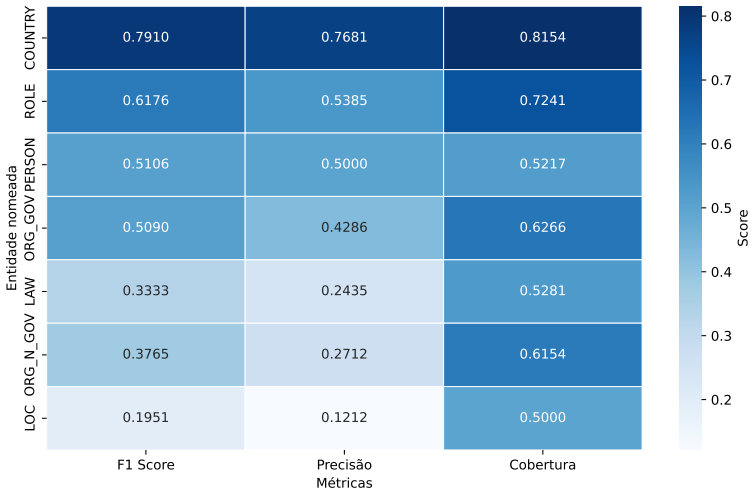

Figura 5.6: Métricas de desempenho do modelo BiLSTM por entidade para combinações de limites exata, ordenados por maior F_1 score (Fonte: autor).

A figura 5.6 contendo as métricas do modelo BiLSTM mostram uma baixa precisão do modelo para identificar especialmente localizações. Isso indica que o modelo está classificando incorretamente palavras que não são referentes a localização como sendo desse tipo. Levando em consideração que entidades nomeadas de localização não possuem a menor quantidade de exemplos do conjunto de dados utilizado, o resultado apresentado pode ter sido afetado por sobreajuste (*overfitting*).

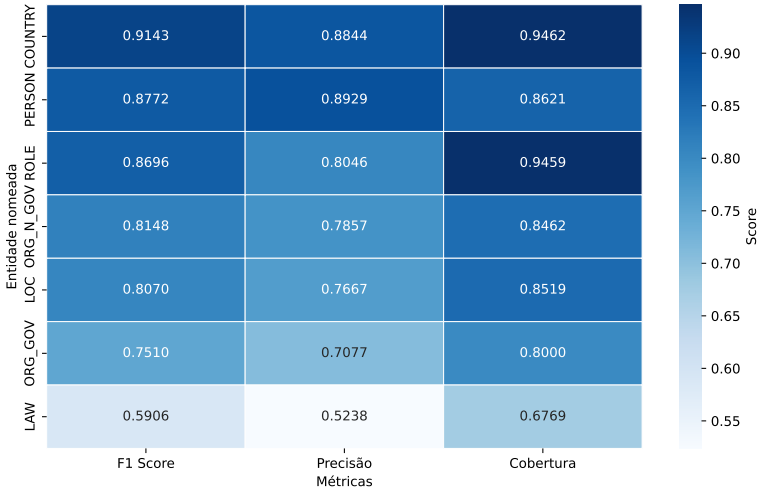

Figura 5.7: Métricas de desempenho do modelo BERT por entidade para combinações de limites exata, ordenados por maior F_1 score (Fonte: autor).

A figura 5.7 mostra uma maior dificuldade do modelo BERT em reconhecer entidades

nomeadas do tipo LAW (leis, tratados e jurisprudências). Essa entidade é caracterizada principalmente por conjuntos maiores de *tokens*. Tendo em vista que as entidades desse tipo podem conter palavras que em outros contextos não são rotulados, uma maior dificuldade dos modelos BiLSTM e BERT para esse tipo de entidade já era esperada. Vale destacar que esse modelo apresenta resultados consideravelmente melhores que os demais para todos os tipos de entidades nomeadas.

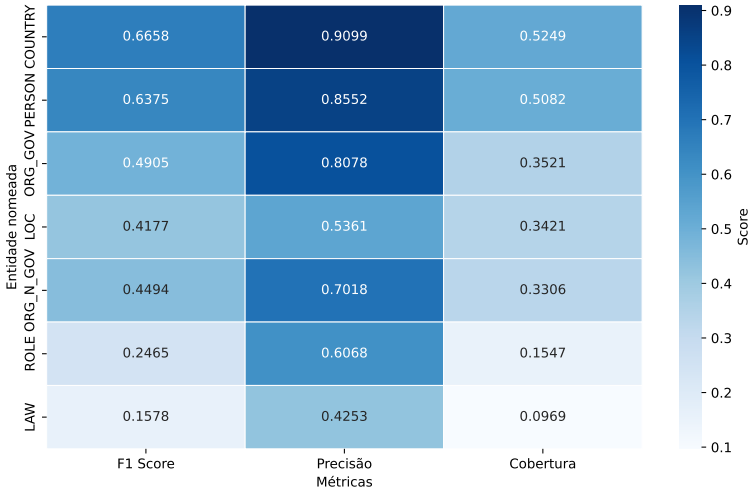

Figura 5.8: Métricas de desempenho do modelo ChatGPT por entidade para combinações de limites exata, ordenados por maior F_1 score (Fonte: autor).

A figura 5.8 mostra as métricas obtidas acerca do reconhecimento de entidades nomeadas a partir do ChatGPT. O modelo, que utilizou apenas 1 exemplo e as instruções para desempenhar a tarefa, apresenta um desempenho melhor em relação a BiLSTM no reconhecimento de entidades nomedas do tipo ORG_N_GOV (organizações não governamentais) e PERSON (nome de pessoas). Para comparação, enquanto o modelo BiLSTM foi treinado com 721 exemplos de entidades nomeadas do tipo COUNTRY e conseguiu obter um $F1$ score de 79,10%, com um esforço significativamente menor ao usuário, o ChatGPT forneceu um resultado de 66,58% de $F1$ score.

5.3 Análise qualitativa

A análise a seguir tem como objetivo avaliar a qualidade do grafo de relações entre entidades nomeadas enquanto método de extração de informações. A discussão é feita com base nos conceitos apresentados a partir da seção 2.4, tendo essa etapa um grau maior de subjetividade.

Para obtenção do grafo de relações de entre entidades nomeadas com o modelo BERT, foram utilizados 15416 postagens da rede social Stack Exchange. O resultado é um grafo

contendo 14781 nós e 103888 arestas. Já para o ChatGPT, foram utilizados 1362 postagens gerando um grafo contendo 707 nós e 1272 arestas.

5.3.1 Centralidade de nós

A análise de centralidade dos nós do grafo gerado a partir do BERT permitiu identificar as entidades nomeadas de "united states" e "president" como entidades mais relevantes para o grafo, corroborando com a predominância do tema política interna e externa americana como mais frequente dentro da rede social. A centralidade de grau nos sugere que a entidade nomeada "united states" está associada a uma diversidade maior de temas, enquanto a centralidade de intermediação sugere que mesmo quando essa entidade não é o tema central, costuma se fazer presente na discussão. Já no grafo do ChatGPT, é possível notar uma presença de "united states", "russia" e "ukraine", dando indícios de que o tema guerra na Ucrânia está mais presente nesse grafo.

Centralidade de grau

Nó	**Centralidade**
united states_COUNTRY	0.2485
president_ROLE	0.1285
united kingdom_COUNTRY	0.0909
party_ORG_GOV	0.0828
government_ORG_GOV	0.0824

Tabela 5.5: Centralidade de grau para o grafo obtido pelo BERT.

Nó	**Centralidade**
united states_COUNTRY	0.2039
russia_COUNTRY	0.0892
germany_COUNTRY	0.0538
ukraine_COUNTRY	0.0538
iran_COUNTRY	0.0481

Tabela 5.6: Centralidade de grau para o grafo obtido pelo ChatGPT.

Centralidade de intermediação

Nó	Centralidade
united states_COUNTRY	0.1995
president_ROLE	0.0675
united kingdom_COUNTRY	0.05245
european union_ORG_GOV	0.0415
government_ORG_GOV	0.03981

Tabela 5.7: Centralidade de intermediação para o grafo obtido pelo BERT.

Nó	Centralidade
united states_COUNTRY	0.2182
russia_COUNTRY	0.0459
united kingdom_COUNTRY	0.0408
germany_COUNTRY	0.0295
ukraine_COUNTRY	0.0251

Tabela 5.8: Centralidade de intermediação para o grafo obtido pelo ChatGPT.

5.3.2 Representações dos subgrafos

Nas representações abaixo a largura das arestas indica uma maior quantidade de postagens realizando a mesma associação entre entidades. A coloração dos nós é responsável por indicar o tipo de entidade nomeada conforme a legenda de cores disponível na figura 5.9, exceto para as visualizações de comunidades detectadas.

Figura 5.9: Legenda de cores para nós do grafo (Fonte: autor).

Subgrafos de relações mais relevantes

O subgrafo de relações mais fortes é obtido por meio da aplicação do algoritmo 2, conforme descrito na seção 4.3.

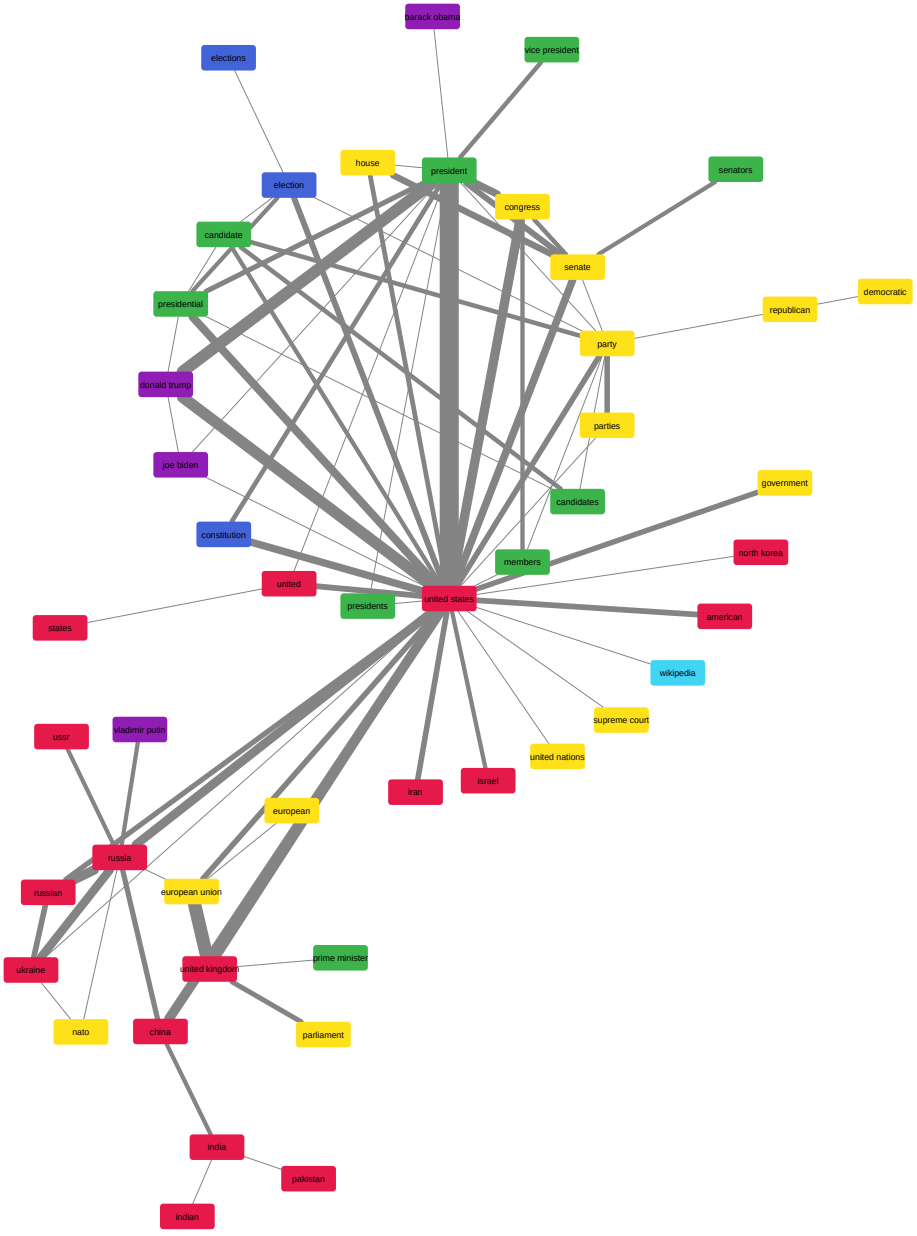

Figura 5.10: Relações mais fortes do grafo obtido utilizando BERT (Fonte: autor).

A figura 5.10[5] demonstra que, quando aplicado o algoritmo 2 ao grafo obtido do modelo BERT, é possível observar um núcleo de relações entre a entidade "united states" e entidades nomeadas associadas a política do país, que se torna também um tema importante para o termo "united kingdom". É possível observar que o subgrafo foi capaz de representar o interesse dos usuários pelos recentes desentendimentos entre a Índia e países vizinhos como o Paquistão e a China[6], assim como a relação dos Estados Unidos com países como Israel, Irã e Coréia do Norte.

[5] `https://github.com/yuridrcosta/tcc/blob/main/img/bert/relacoes_mais_relevantes_bert.pdf`

[6] `https://pt.wikipedia.org/wiki/Conflito_na_Caxemira`

Os presidentes dos Estados Unidos Barack Obama, Donald Trump e Joe Biden, junto com o presidente da Rússia Vladimir Putin ganham destaque entre as pessoas mais comentadas da rede social.

4.3.

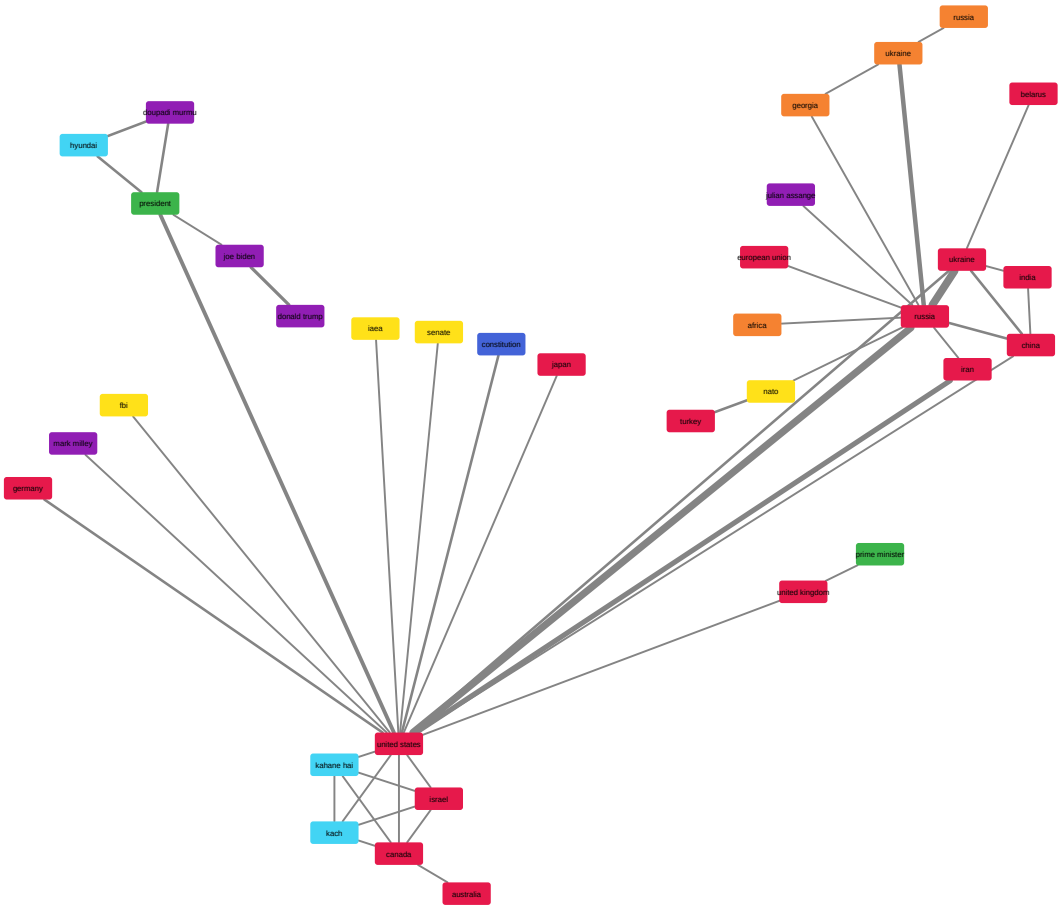

Figura 5.11: Relações mais fortes do grafo obtido utilizando ChatGPT (Fonte: autor).

A figura 5.11[7] mostra o subgrafo obtido a partir da aplicação do algoritmo 2 ao grafo obtido do ChatGPT e nele é possível perceber alguns equívocos do modelo entre entidades nomeadas COUNTRY e LOC. Apesar de classificar corretamente África enquanto localização, a definição da entidade nomeada COUNTRY englobava o nome de todos os países, logo "georgia", "ukraine" e "russia" foram classificados incorretamente.

A parte superior direita da figura 5.11 foi capaz de explicitar diversas relações da política externa de países como Rússia, China, India e Ucrânia, com destaque para o papel da Turquia enquanto país-membro da OTAN com poder de veto, tendo essa relação Rússia-Otan-Turquia muita relevância no contexto da guerra entre Rússia e Ucrânia[8].

[7] `https://github.com/yuridrcosta/tcc/blob/main/img/chatgpt/relacoes_mais_relevantes_chatgpt.pdf`

[8] `https://www.bbc.com/portuguese/internacional-61474467`

Subgrafos de conexões diretas para entidades nomeada

As visualizações a seguir foram obtidas a partir da aplicação do algoritmo 2 a um nó central desejado, conforme descrito na seção 4.3.2. O peso mínimo das arestas foi ajustado conforme a quantidade de ligações do nó central de modo a manter na visualização uma quantidade de nós entre 10 e 25. Foi utilizado o layout circular para obtenção das representações.

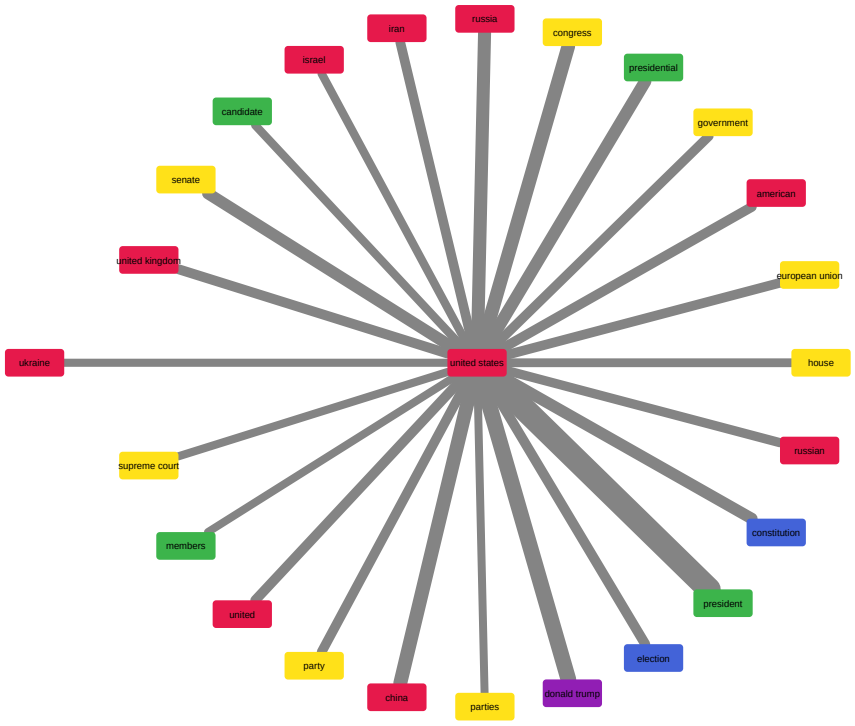

Figura 5.12: Conexões mais relevantes com United States para o modelo BERT (Fonte: autor).

A figura 5.12[9], que representa as conexões mais recorrentes da entidade "united states", é capaz de evidenciar um maior interesse dos usuários em comentar a política interna americana, especialmente eleições, em comparação com a política externa. Dentre a pessoa mais comentada está o ex-presidente Donald Trump, que é reconhecido como figura polêmica e midiática daquele país.

[9] `https://github.com/yuridrcosta/tcc/blob/main/img/bert/usa_connections.pdf`

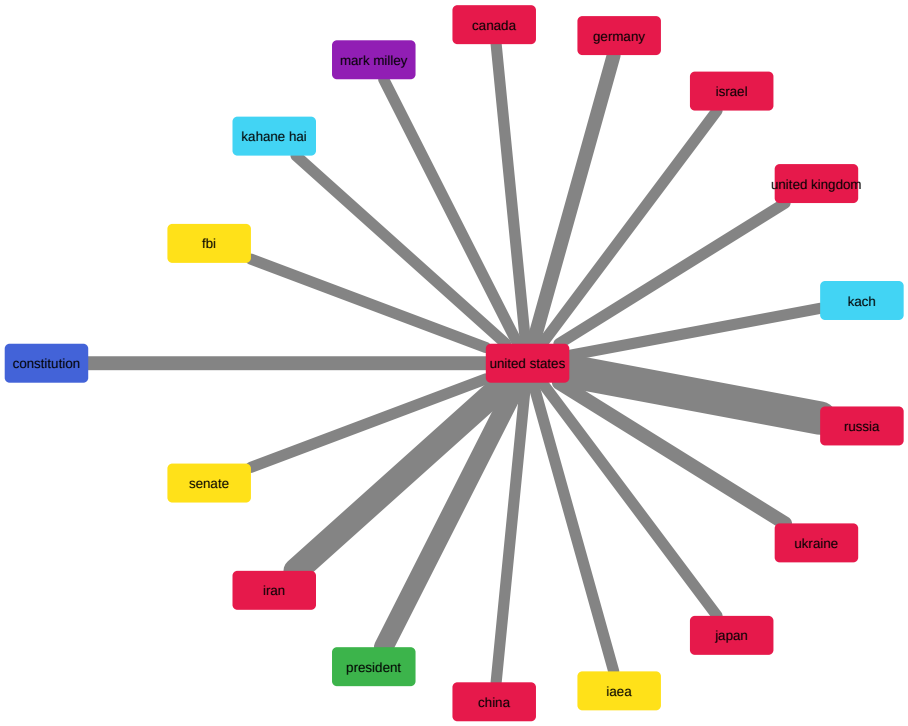

Figura 5.13: Conexões mais relevantes com United States para o modelo ChatGPT (Fonte: autor).

Já as principais conexões para os Estados Unidos de acordo com o grafo obtido do ChatGPT (figura 5.13[10]) traz como principal diferença a associação em postagens do país norte americano com o partido Kach israelense. Essa relação traz uma informação de maior dificuldade de acesso, tendo em vista que se trata de um partido político considerado extremista por alguns jornalistas[11]. As entidades "kahane hai" e "kach" não estão presentes no grafo do BERT, apesar de terem sido manualmente rotuladas como ORG_N_GOV (nome de organizações não governamentais ou empresas) e ORG_GOV (nomes de orgãos governamentais, partidos políticos ou empresas públicas), respectivamente.

[10] `https://github.com/yuridrcosta/tcc/blob/main/img/chatgpt/usa_connections.pdf`
[11] `https://www.bbc.com/portuguese/internacional-47825686`

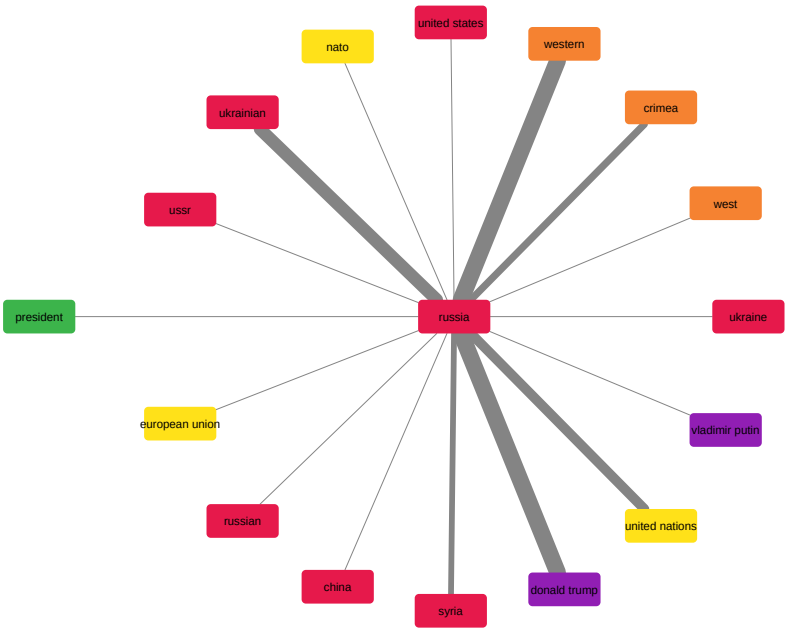

Figura 5.14: Conexões mais relevantes com Russia para o modelo BERT (Fonte: autor).

Á figura 5.14[12] evidencia que uma grande parte dos comentários dos usuários da rede social acerca da Rússia estão relacionados a relação desse país com o ex-presidente americano Donald Trump[13], corroborando também com a evidência de que a maior parte das interações na rede falam sobre os Estados Unidos. O conflito do país com a Ucrânia[14] fica evidenciado pela aparição das entidades "ukrainian" e "crimea".

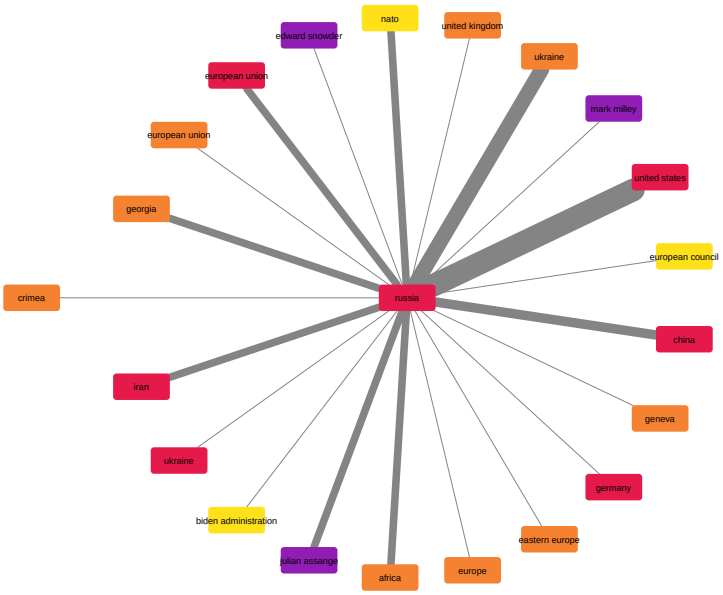

Figura 5.15: Conexões mais relevantes com Rússia para o modelo ChatGPT (Fonte: autor).

[12] https://github.com/yuridrcosta/tcc/blob/main/img/bert/russia_connections.pdf
[13] https://www.theguardian.com/us-news/trump-russia-inquiry
[14] https://pt.wikipedia.org/wiki/Guerra_Russo-Ucraniana

A figura 5.15[15] mostra as principais conexões da Rússia de acordo com o ChatGPT. É possível observar como principal diferença em relação ao BERT a presença das entidades de Edward Snowden[16] e Julian Assange. Essas entidades nomeadas, no entanto, estão presentes também no grafo do BERT, e não aparecerem devido ao grafo do BERT ter um maior número de postagens, não tendo essas entidades figurado entre as principais. O subgrafo do ChatGPT também ilustra uma predominância da classificação de Ucrânia enquanto localização, classificação que apesar de válida no contexto puramente de localização geográfica, não é a mais adequada tendo em vista que a maioria das postagens mencionando o país se referem a guerra entre os países Ucrânia e Rússia.

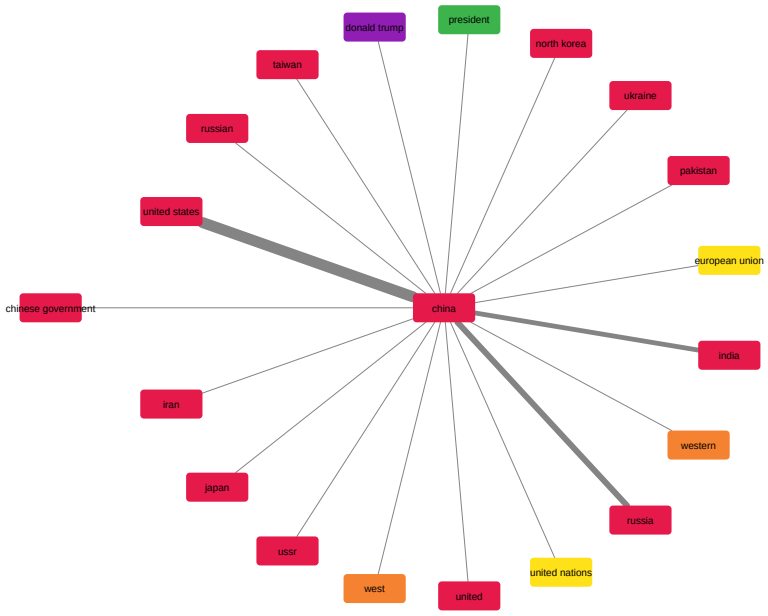

Figura 5.16: Conexões mais relevantes com China para o modelo BERT (Fonte: autor).

A figura 5.16[17] traz grande destaque as questões geopolíticas as quais a China está envolvida, em especial sua relação com a entidade de Estados Unidos evidencia o interesse pela guerra econômica entre esses países. É possível observar que o interesse dos usuários que comentam acerca da China está focado na sua relações com demais países e não na política interna como evidenciado para a entidade "united states".

[15] https://github.com/yuridrcosta/tcc/blob/main/img/chatgpt/russia_connections.pdf

[16] https://g1.globo.com/mundo/noticia/2022/09/26/putin-da-cidadania-russa-a-edward-snowden.ghtml

[17] https://github.com/yuridrcosta/tcc/blob/main/img/bert/china_connections.pdf

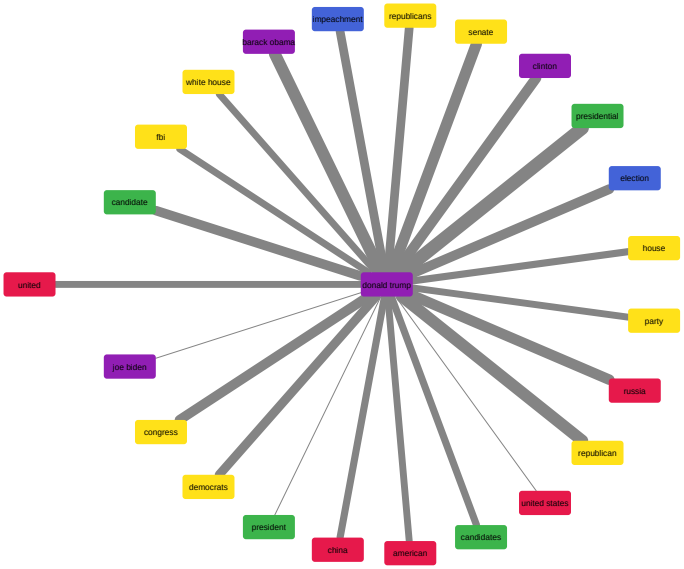

Figura 5.17: Conexões mais relevantes com Donald Trump para o modelo BERT (Fonte: autor).

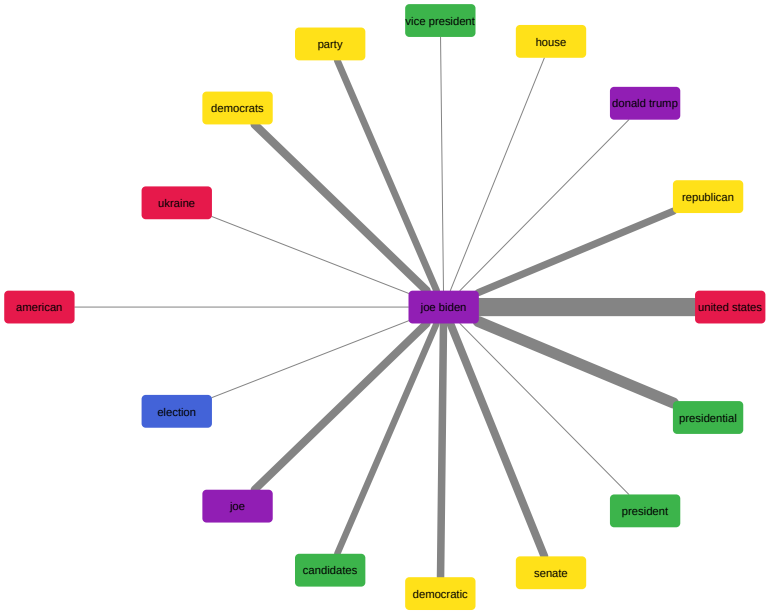

Figura 5.18: Conexões mais relevantes com Joe Biden para o modelo BERT (Fonte: autor).

Dentre as entidades nomeadas de pessoas selecionadas, a figura 5.17[18] é capaz de evidenciar o interesse dos usuários da rede social pela suposta ligação entre Donald Trump e a Rússia. Vale destacar a relação mais forte entre Trump e o seu antecessor Barack Obama do que com Hillary Clinton que concorreu às eleições americanas com Trump, podendo indicar uma maior ocorrência de comparações entre governo na rede social do que comentários sobre a campanha eleitoral. Já o subgrafo de Joe Biden (figura 5.18 [19]) está mais associado a sua posição enquanto político de longa carreira[20], tendo como menções principais, além do próprio país, o senado, ao qual frequentou durante anos.

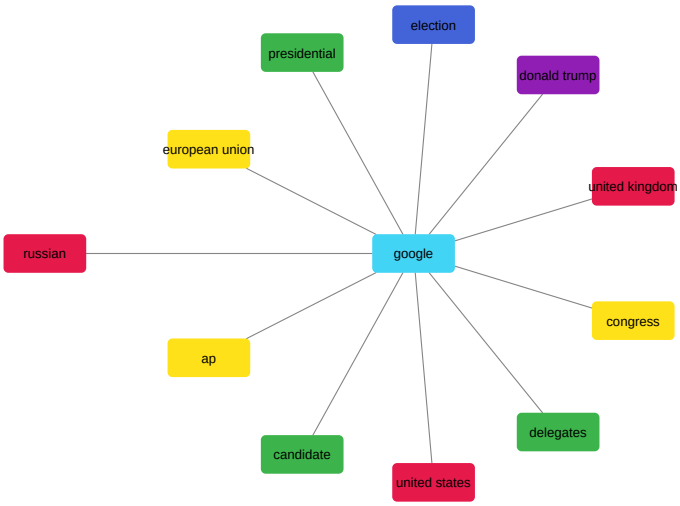

Figura 5.19: Conexões mais relevantes com Google para o modelo BERT (Fonte: autor).

Para a Google, a figura 5.19 trouxe conexões que podem ter surgido de temas variados, com um destaque maior para as eleições americannas. Além disso, a conexão com a entidade nomeada da União Europeia ("european union") destaca o recente processo jurídico relacionado a leis anti-monopólio.

[18] `https://github.com/yuridrcosta/tcc/blob/main/img/bert/trump_connections.pdf`
[19] `https://github.com/yuridrcosta/tcc/blob/main/img/bert/joe_biden_connections.pdf`
[20] `https://pt.wikipedia.org/wiki/Joe_Biden`

5.3.3 Detecção de comunidades

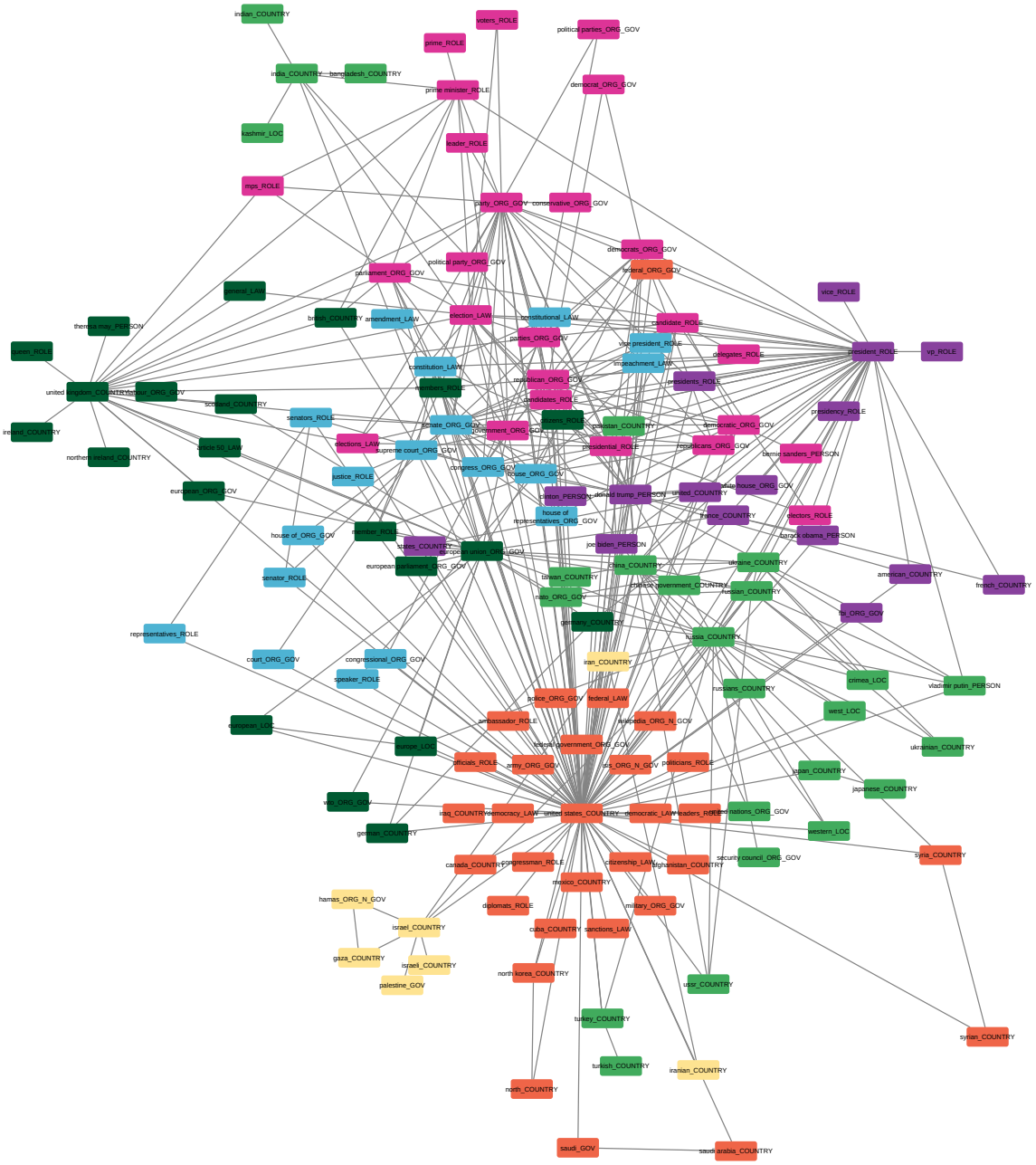

Figura 5.20: Grafo de comunidades detectadas para o modelo BERT (Fonte: autor).

A figura 5.20[21] mostra o subgrafo do BERT com entidades que apareceram relacionadas em pelo menos 15 postagens onde cada cor representa uma comunidade, totalizando 6 comunidades detectadas em um grafo contendo 142 nós e 385 arestas. A modularidade das comunidades foi de $0,4027$.

O algoritmo mostrou uma boa separação para entidades relacionadas ao Europa, ficando todas em uma comunidade representada pela cor verde escuro. Entidades nomeadas

[21] `https://github.com/yuridrcosta/tcc/blob/main/img/bert/comunidades_bert.pdf`

relacionadas a política russa se concentraram na cor verde claro e entidades nomeadas relacionadas a política israelense ficaram na comunidade de cor amarela. A cor laranja mostrou uma predominância da política externa americana. No entanto, é possível observar que entidades relacionadas a política interna americana foram divididas em 3 comunidades diferentes representadas pelas cores rosa, azul claro e roxo. Não foi possível identificar motivação aparente para essa separação.

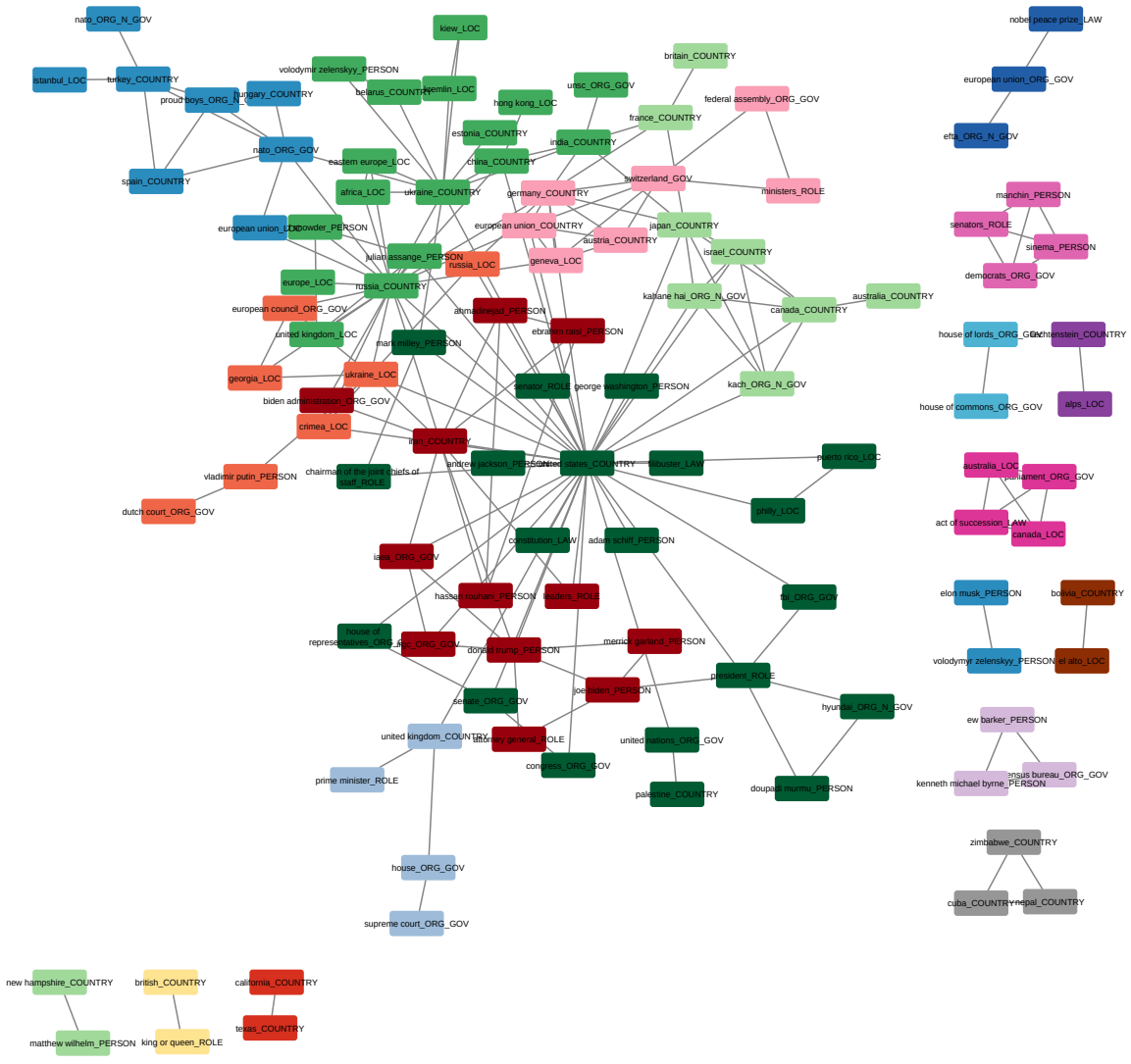

Figura 5.21: Grafo de comunidades detectadas para o modelo ChatGPT (Fonte: autor).

A figura 5.21[22] traz o subgrafo de comunidades do ChatGPT. Foram mantidas no subgrafo as arestas entre entidades nomeadas que apareceram juntas em pelo menos 2 postagens. O algoritmo de Louvain gerou um total de 19 comunidades para os 114 nós e

[22] `https://github.com/yuridrcosta/tcc/blob/main/img/chatgpt/comunidades_chatgpt.pdf`

as 184 arestas do grafo utilizado. Apesar de 19 comunidades terem sido identificadas, 12 delas são de grafos desconexos, o que, pelo algoritmo, sempre gera uma nova comunidade. A modularidade das comunidades foi de $0,6580$.

O algoritmo mostra uma separação mais significativa entre as comunidades em geral. Enquanto no subgrafo do BERT, o tema política americana foi muito presente e acabou se dividindo entre 3 comunidades, para o subgrafo do ChatGPT os temas se mostraram mais variados e as comunidades melhor definidas. O algoritmo foi capaz de identificar uma comunidade mais associada ao tema Irã, representada em marrom. Além disso, assuntos relacionados aos Estados Unidos ficaram predominantemente em uma única comunidade, de cor verde escuro. Uma comunidade relacionada a união europeia pode ser observada representada na cor de rosa mais claro.

Conclusão

Este trabalho apresentou um método para a extração de informações em grandes volumes de dados textuais por meio da estruturação em redes complexas a partir do reconhecimento de entidades nomeadas. A comparação com o modelo BiLSTM e BERT permitiu demonstrar a viabilidade da realização do reconhecimento de entidades nomeadas utilizando *prompt engineering* em modelos *Large Language Models*, como o ChatGPT. Embora o método apresente limitações, conforme discutido, foi possível, a partir de postagens de usuários da rede social Stack Exchange sobre política, construir uma representação em grafo que captura informações sutis a partir do conteúdo disponível.

O estudo demonstra que, ainda que tenha se mostrado distante do estado da arte para o conjunto de dados, o modelo ChatGPT mostrou-se eficiente na tarefa de reconhecimento de entidades nomeadas, especialmente considerando o uso de apenas um exemplo como entrada. Desse modo, o modelo se mostrou capaz de reduzir significativamente o esforço manual, sendo uma opção viável para contextos onde não haja uma necessidade de uma cobertura alta. Por meio do grafo de relações entre entidades nomeadas obtido foi possível comprovar a utilidade enquanto método de extração automática de informações, sendo possível observar relações que exigiriam um maior esforço para a obtenção da informação de maneira manual na rede social.

A utilização do modelo BERT refinado com exemplos rotulados manualmente revelou-se a abordagem mais robusta e confiável para aplicação do método. Esse modelo se mostrou mais adequado quando observado as métricas de precisão e cobertura, que se mostraram superiores em todos os tipos de entidades nomeadas dispostas no conjunto de dados. Além disso, por se tratar de um modelo com pesos pré-treinados disponíveis foi possível analisar uma quantidade maior de documentos sem custos variáveis por quantidade de tokens, como ocorre ao modelo ChatGPT.

Dentre as limitações do presente trabalho, destaca-se a quantidade de textos aplicados ao ChatGPT. Tal limitação ocorreu devido aos limites de tempo e quantidade de tokens por usuário impostos pela empresa proprietária do serviço. O número menor de postagens afetou análise qualitativa do método por meio da rede complexa obtida das postagens da rede social, não sendo suficiente para analisar por completo os padrões de seus usuários. No entanto, essa limitação não afeta a análise quantitativa tendo em vista que, por ter sido fornecido apenas um exemplo ao modelo, todo o conjunto de dados foi utilizado para

a obtenção das métricas e são capazes de indicar a baixa cobertura ao utilizar o modelo na tarefa de reconhecimento de entidades nomeadas.

Para trabalhos futuros, sugere-se a aplicação da metodologia proposta a diferentes conjuntos de dados e o aumento da quantidade de documentos submetidos ao ChatGPT. Além disso, a exploração de outras técnicas de análise de grafos pode tornar a abordagem mais eficiente na análise de grandes volumes de dados textuais.

Bibliografia

Affi, M. and Latiri, C. (2021). Be-blc: Bert-elmo-based deep neural network architecture for english named entity recognition task. *Procedia Computer Science*, 192:168–181. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 25th International Conference KES2021.

Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. (1998). Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Sixth Workshop on Very Large Corpora*.

Chinchor, N. and Robinson, P. (1998). Appendix E: MUC-7 named entity task definition (version 3.5). In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

Chinchor, N. and Sundheim, B. (1993). MUC-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.

Cho, M., Ha, J., Park, C., and Park, S. (2020). Combinatorial feature embedding based on cnn and lstm for biomedical named entity recognition. *Journal of Biomedical Informatics*, 103:103381.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Esuli, A. and Sebastiani, F. (2010). Evaluating information extraction. In *CLEF*, pages 100–111.

Exchange, P. S. (2024). Politics stack exchange. `https://politics.stackexchange.com`. Accessed: 2024-07-08.

Fan, R., Wang, L., Yan, J., Song, W., Zhu, Y., and Chen, X. (2020). Deep learning-based named entity recognition and knowledge graph construction for geological hazards. *ISPRS International Journal of Geo-Information*, 9(1).

Fruchterman, T. M. and Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164.

Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826.

Golbeck, J. (2015). Chapter 21 - analyzing networks. In Golbeck, J., editor, *Introduction to Social Media Investigation*, pages 221–235. Syngress, Boston.

Goodfellow, I. J., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge, MA, USA. `http://www.deeplearningbook.org`.

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J., editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA.

Hirschberg, J. and Manning, C. D. (2015). Advances in natural language processing. *Science (American Association for the Advancement of Science)*, 349(6245):261–266.

Hochreiter, S. and Schmidhuber, J. (1997a). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hochreiter, S. and Schmidhuber, J. (1997b). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Korkontzelos, I., Piliouras, D., Dowsey, A. W., and Ananiadou, S. (2015). Boosting drug named entity recognition using an aggregate classifier. *Artificial Intelligence in Medicine*, 65(2):145–153. Intelligent healthcare informatics in big data era.

Li, J., Sun, A., Han, J., and Li, C. (2022). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Liu, X. and Zhou, M. (2013). Two-stage ner for tweets with clustering. *Information Processing Management*, 49(1):264–273.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization.

Meteo Service (2022). Meteo data. This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit `http://creativecommons.org/licenses/by/4.0/`.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mitchell, T. M. (1997). *Machine learning*, volume 1. McGraw-hill New York.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. (2024). A comprehensive overview of large language models.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M.,

McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. (2024). Gpt-4 technical report.

Patil, N., Patil, A., and Pawar, B. (2020). Named entity recognition using conditional random fields. *Procedia Computer Science*, 167:1181–1188. International Conference on Computational Intelligence and Data Science.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations.

Radford, A. and Narasimhan, K. (2018). Improving language understanding by generative pre-training.

Ribeiro, H. V., Alves, L. G. A., Martins, A. F., Lenzi, E. K., and Perc, M. (2018). The dynamical structure of political corruption networks. *Journal of Complex Networks*, 6(6):989–1003.

Russe, M., Reisert, M., and Rau, A. (2024). Improving the use of llms in radiology through prompt engineering: from precision prompts to zero-shot learning. *RöFo - Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren*.

Sang, E. F. T. K. and Meulder, F. D. (2003). Introduction to the CoNLL-2003 shared task: Language-Independent Named Entity Recognition. In Daelemans, W. and Osborne,

M., editors, *Proceedings of CoNLL-2003 and the 7th Conference on Natural Language Learning*, pages 142–147.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2020). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Sawicki, J., Ganzha, M., Paprzycki, M., and Watanobe, Y. (2024). Applying named entity recognition and graph networks to extract common interests from thematic subfora on reddit. *Applied sciences*, 14(5):1696.

Segura-Bedmar, I., Martínez, P., and Herrero-Zazo, M. (2013). SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In Manandhar, S. and Yuret, D., editors, *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Shenfield, A. and Howarth, M. (2020). A novel deep learning model for the detection and identification of rolling element-bearing faults. *Sensors (Basel, Switzerland)*, 20.

Singh, T. D., Nongmeikapam, K., Ekbal, A., and Bandyopadhyay, S. (2009). Named entity recognition for manipuri using support vector machine. volume 2.

Six, J. M. and Tollis, I. G. (1999). Circular drawing algorithms. In Tamassia, R., editor, *Handbook of Graph Drawing and Visualization*, pages 285–315. CRC Press.

Talagala, N. (2022). Data as the new oil is not enough: Four principles for avoiding data fires. Accessed: 2024-10-05.

Tkachenko, M., Malyuk, M., Holmanyuk, A., and Liubimov, N. (2020-2022). Label Studio: Data labeling software. Open source software available from https://github.com/heartexlabs/label-studio.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., Li, J., and Wang, G. (2023). Gpt-ner: Named entity recognition via large language models.

Wei, S., Liang, Y., Li, X., Weng, X., Fu, J., and Han, X. (2023). Chinese few-shot named entity recognition and knowledge graph construction in managed pressure drilling domain. *Entropy (Basel, Switzerland)*, 25(7):1097.

Zhou, G. and Su, J. (2002). Named entity recognition using an HMM-based chunk tagger. In Isabelle, P., Charniak, E., and Lin, D., editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 473–480, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Zorzi, M., Testolin, A., and Stoianov, I. P. (2013). Modeling language and cognition with deep unsupervised learning: a tutorial overview. *Frontiers in psychology*, 4:515–515.