Modelos Profundos de Linguagem para Reconhecimento de Entidades Nomeadas em Domínio Jurídico

Luiz Henrique Neves Bonifacio

Modelos Profundos de Linguagem para Reconhecimento de Entidades Nomeadas em Domínio Jurídico[1]

Luiz Henrique Neves Bonifacio

Orientador: *Profº Drº Eraldo Luís Rezende Fernandes*

Dissertação entregue a Faculdade de Computação da Universidade Federal de Mato Grosso do Sul - FACOM-UFMS - como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

UFMS - Campo Grande
Maio/2020

[1]Trabalho Realizado com Auxílio da CAPES Proc. No: 88882.458433/2019-01

Se aprendesse qualquer coisa,
necessitaria aprender mais, e
nunca ficaria satisfeito.

Graciliano Ramos, *Vidas Secas*

Agradecimentos

Assim como tenho feito ao longo de todos estes anos, expresso toda minha gratidão e reconhecimento à minha mãe, Dilma Neves, que jamais mediu esforços para garantir que nada me faltasse e que é meu apoio inabalável em todos os momentos da minha vida. Professora da rede pública de ensino, ela me ensina todos os dias que o amor e a dedicação à educação são capazes de realizar transformações únicas. Agradeço também à minha irmã, Letícia, que mesmo com minha distância durante todos estes anos, compreende as minhas escolhas e se mantém presente.

Vivemos tempos diferentes. A pandemia causada pelo coronavírus escancarou uma necessidade há muito conhecida e negligenciada: a ciência. Numa sociedade onde evidências científicas são refutadas por opiniões infundadas, é necessário reconhecer aqueles que fazem da ciência um lugar seguro. Agradeço imensamente ao Profº Drº Eraldo Fernandes, por toda orientação e dedicação durante os últimos dois anos. Obrigado por essa jornada, por me guiar em um caminho no qual eu pouco conhecia e por todas as correções (sempre muito firmes e cirúrgicas). Considero que meu aprendizado não seria o mesmo sem o seu exemplo. Também agradeço todos os professores da Faculdade de Computação da UFMS por toda a dedicação aos cursos de graduação e principalmente ao Programa de Pós-Graduação. Em um cenário de completa desvalorização, todos se apresentam disponíveis para que o programa continue.

Não poderia deixar de agradecer meu grupo de amigos, que se tornou minha segunda família: Bruna, Guilherme, Laura e Natyelle. Vocês tornaram essa jornada muito mais leve, me ouvindo e me apoiando nos momentos difíceis, comemorando minhas vitórias e me consolando nas minhas derrotas. Sou eternamente grato pela amizade de cada um de vocês e por esse grupo que não se desfaz de modo algum.

Resumo

Modelos profundos de linguagem, como ELMo, BERT e GPT, alcançaram resultados impressionantes em várias tarefas de linguagem natural. Tais modelos são pré-treinados em grandes corpora construídos a partir de textos de domínio geral, sem qualquer tipo de anotação, e posteriormente treinados de forma supervisionada em uma tarefa final. Uma etapa opcional consiste em realizar um ajuste fino no modelo de linguagem utilizando um corpus intradomínio que seja suficientemente grande e sem anotações, antes de treinar o modelo na tarefa de interesse. Esta abordagem não é amplamente explorada na literatura atual. Neste trabalho, é investigado o impacto causado por esta etapa por meio da tarefa de reconhecimento de entidades nomeadas (REN) em documentos jurídicos no idioma Português. São explorados diferentes cenários, considerando duas arquiteturas de modelos de linguagem baseadas em aprendizagem profunda (ELMo e BERT), quatro corpora sem anotações e três tarefas de REN, uma relacionada ao domínio geral e duas pertencentes ao domínio jurídico, todas no idioma Português. Resultados experimentais mostram uma melhora significativa no desempenho devido ao *finetuning* do modelo de linguagem em textos intradomínio. Os modelos treinados também foram avaliados em duas tarefas de REN de domínio geral, com o objetivo de verificar se as melhorias obtidas foram devidas à similaridade entre os domínios ou simplesmente a maior quantidade de dados de treinamento. Os resultados alcançados indicam que realizar *finetuning* em dados do domínio jurídico prejudica o desempenho do modelo quando avaliado em tarefas de REN em dados de domínio geral. Além disso, o modelo de linguagem baseado na arquitetura BERT, com *finetuning* em um corpus do domínio jurídico melhorou significantemente o resultado estado-da-arte para o corpus LeNER-Br, um corpus de REN formado por documentos jurídicos em Português.

Palavras-chave: Processamento de Linguagem Natural, Reconhecimento de Entidades, Aprendizagem Profunda

Abstract

Deep language models, like ELMo, BERT and GPT, have achieved impressive results on several natural language tasks. Such language models are pretrained on large corpora of unlabeled, general domain text and later supervisedly trained on downstream tasks. An optional step consists of finetuning the language model on a large intradomain corpus of unlabeled text, before training it on the final task. This aspect is not well explored in the current literature. In this work, we investigate the impact of this step on named entity recognition (NER) for Portuguese legal documents. We explore different scenarios considering two deep language architectures (ELMo and BERT), four unlabeled corpora and three legal NER tasks for the Portuguese language. Experimental findings show a significant improvement on performance due to language model finetuning on intradomain text. We also evaluate the trained models on two general-domain NER tasks, in order to understand whether the aforementioned improvements were really due to domain similarity or simply due to more training data. The achieved results indicate that finetuning on a legal domain corpus hurts performance on the general-domain NER tasks. Additionally, our BERT model, finetuned on a legal corpus, significantly improves on the state-of-the-art performance on the LeNER-Br corpus, a Portuguese language NER corpus for the legal domain.

Keywords: Natural Language Processing, Named Entity Recognition, Deep Learning

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AM Aprendizado de Máquina

AP Aprendizado Profundo

BERT Bidirectional Encoder Representations from Transformers

biLM bidirectional Language Model

CNJ Conselho Nacional de Justiça

CNN Convolutional Neural Network

DL Deep Learning

ELMo Embeddings from Language Models

LSTM Long Short-Term Memory

ML Modelo de Linguagem

MLM Modelo de Linguagem Mascarado

MUC-6 Sixth Message Understanding Conference

PLN Processamento de Linguagem Natural

PPS Predição de Próxima Sentença

REN Reconhecimento de Entidades Nomeadas

ULMFiT Universal Language Model Fine-tuning

Conteúdo

CAPÍTULO

1

Introdução

A técnica de transferência de aprendizagem consiste em utilizar o conhecimento adquirido por um modelo treinado em determinado conjunto de dados e aplicá-lo em outro conjunto distinto [Mou et al., 2016]. Na área de Processamento de Linguagem Natural (PLN) este processo é realizado por meio do pré-treinamento de Modelos de Linguagem (MLs) e tem atingido resultados expressivos em diversas tarefas. Estes MLs, muitas vezes baseados em aprendizagem profunda, são pré-treinados em grandes quantidades de origem diversa. Modelos como ULMFiT [Howard and Ruder, 2018], ELMo [Peters et al., 2018], BERT [Devlin et al., 2018] e GPT [Radford et al., 2018] são exemplos de modelos profundos de linguagem, capazes de gerar representações poderosas para as palavras de um texto. Comumente, as representações são incorporadas a modelos que são treinados de maneira supervisionada para resolver alguma tarefa específica.

Embora MLs pré-treinados em domínio geral tenham alcançado sucesso em muitas tarefas, alguns domínios apresentam características particulares que não são facilmente identificadas. Este aspecto pode limitar o desempenho destes modelos quando utilizados para resolver tarefas dentro de domínios específicos. Nestes casos, uma alternativa é realizar o *finetuning* destes MLs, utilizando dados pertencentes ao domínio de interesse. O processo consiste em realizar um ajuste fino do ML, por meio de um corpus construído por documentos do domínio de interesse. Dessa forma, deseja-se garantir que o ML possa se ajustar as características específicas da linguagem daquele domínio. Este processo deve ocorrer após o pré-treinamento do ML e antes do treinamento supervisionado na tarefa final. A ideia foi explorada com sucesso em trabalhos anteriores [Howard and Ruder, 2018, Radford et al., 2018, Lample

and Conneau, 2019] mas ainda é incipiente, particularmente para português.

O domínio jurídico apresenta um cenário onde existe uma enorme quantidade de texto puro, mas há escassez de dados anotados, especialmente para idiomas como o Português. Neste trabalho, é explorado o impacto do *finetuning* de MLs profundos em documentos jurídicos antes do treinamento supervisionado para a tarefa de Reconhecimento de Entidades Nomeadas (REN), que consiste em localizar e identificar as entidades nomeadas presentes em um texto. Foram utilizadas as arquiteturas BERT e ELMo, cujos MLs foram treinados em três corpora de domínio geral: 100 maiores Wikipédias (BERT Multilingual), Wikipédia em Português e brWaC (corpus em Português formado por 2.7bi de tokens). Os modelos de linguagem de domínio geral foram submetidos ao processo de *finetuning* no corpus de domínio jurídico denominado Acórdãos-TCU[1]. Todos os MLs foram avaliados por meio da tarefa de Reconhecimento de Entidades Nomeadas (REN) considerando os corpora HAREM [Cardoso, 2006, Santos and Cardoso, 2007], LeNER-Br [Luz de Araujo et al., 2018] e DrugSeizures-Br. De forma breve, cada corpus é descrito da seguinte maneira:

(a) HAREM: documentos obtidos de fontes diversas e anotados com entidades de domínio geral.

(b) LeNER-Br: documentos obtidos de diferentes tribunais brasileiros juntamente com quatro documentos legislativos, todos com anotações de entidades gerais e jurídicas.

(c) DrugSeizures-Br: corpus privado formado por petições relacionadas ao tráfico de drogas protocoladas junto ao Ministério Público de Mato Grosso do Sul.

Os dados utilizados no último cenário foram coletados durante o desenvolvimento deste trabalho.

Resultados experimentais mostraram que modelos de linguagem treinados em dados do domínio jurídico apresentam desempenho superior em comparação aos modelos de domínio geral no cenário (b), enquanto prejudicam o desempenho no cenário (a). Estes resultados indicam que o*finetuning* de modelos de linguagem em dados do domínio de interesse é benéfico. Contudo, as avaliações do cenário (c) mostraram comportamento distinto. DrugSeizures-Bré um corpus formado por documentos jurídicos, porém seu estilo de linguagem é diferente dos estilos encontrados nos corpora LeNER-Br e Acórdãos-TCU. Neste caso, não foi possível encontrar melhoras significativas no desempenho ao utilizar modelos treinados em dados jurídicos ao invés de dados de domínio

[1]https://github.com/netoferraz/acordaos-tcu

geral. De fato, em alguns casos, o *finetuning* em dados de domínio específico prejudica o desempenho dos modelos. Isso significa que o *finetuning* intradomínio é favorável quando realizado em um corpus que seja suficientemente similar ao corpus da tarefa de interesse. Características como o tamanho do corpus, a variabilidade linguística e o estilo de escrita adotado podem causar influência neste processo. Em outras palavras, embora os corpora pertençam ao mesmo domínio, eles podem apresentar diferenças consideráveis. Tais diferenças podem resultar em impactos negativos quando considerado o processo de *finetuning* de um ML, e posterior avaliação na tarefa final. Adicionalmente, os experimentos realizados durante a execução deste trabalho alcançaram novos resultados estado-da-arte para o corpus LeNER-Br.

1.1 Objetivos

O objetivo geral deste trabalho é verificar o impacto do processo de *finetuning* de modelos profundos de linguagem, utilizando dados de domínio específico, avaliando estes modelos por meio da tarefa de REN. Para que o objetivo geral seja alcançado, são propostos os seguintes objetivos específicos:

- Investigar se o processo de *finetuning* intradomínio, considerando o domínio jurídico, é capaz de melhorar o desempenho dos MLs avaliados em tarefas de REN do mesmo domínio.

- Investigar qual o efeito do processo de *finetuning* intradomínio quando os MLs são avaliados em tarefa de REN de domínio geral.

- Investigar o impacto do processo de *finetuning* de MLs em corpus de domínio geral, mas no idioma Português, quando comparado com ML pré-treinado em diversos idiomas.

- Realizar o processo de *finetuning* de MLs de duas arquiteturas distintas, utilizando dados do domínio jurídico.

- Avaliar o impacto do *finetuning* nos MLs por meio dos corpora de REN propostos na metodologia deste trabalho.

- Comparar modelos de linguagem pré-treinados de forma multi-idioma e mono-idioma.

- Estabelecer comparações em relação às arquiteturas BERT e ELMo, quando avaliadas na tarefa de REN no corpus LeNER-Br.

1.2 Contribuições

As principais contribuições deste trabalho podem ser resumidas em:

- Modelos profundos de linguagem treinados em dados do domínio jurídico.
- Avaliação da técnica de finetuning intra-domínio para REN em domínio jurídico.
- Novo modelo com desempenho estado-da-arte na avaliação do corpus LeNER-Br.

1.3 Resumo dos Resultados

Por meio das avaliações realizadas neste trabalho, foram estabelecidas as conclusões:

- O impacto do *finetuning* de modelos de linguagem profundos está relacionado a características do corpus utilizado para o *finetuning*, bem como as características do corpus utilizado para avaliação da tarefa de REN.
- A etapa de pré-treinamento do modelo de linguagem tem papel importante quando considerado o desempenho do modelo quando avaliado na tarefa de REN.
- Diferenças arquiteturais entre os modelos de linguagem também influenciam no desempenho dos modelos, quando avaliados em tarefas de REN, independente do domínio.

A estrutura deste trabalho está organizada da seguinte maneira: No Capítulo 2, são descritos os conceitos, técnicas e modelos profundos de linguagem, que compõem o referencial teórico deste trabalho. No Capítulo 3, a metodologia do trabalho é descrita em detalhes, assim como os corpora utilizados para o desenvolvimento experimental. No Capítulo 4, são discutidos os principais trabalhos relacionados envolvendo à tarefa de REN, o uso de modelos de linguagem, bem como suas etapas de pré-treinamento e *finetuning*. Os resultados experimentais são apresentados e discutidos no Capítulo 5. Por fim, no Capítulo 6, são apresentadas as principais conclusões, as limitações e riscos deste trabalho e possíveis trabalhos futuros.

CAPÍTULO

2

Referencial Teórico

A tarefa de Reconhecimento de Entidades Nomeadas desempenha um papel importante dentro da área de Processamento de Linguagem Natural. Além de ser uma tarefa final, uma aplicação por si só, ela também é parte fundamental na realização de outras tarefas como Recuperação de Informação, por exemplo. A utilização de modelos de linguagem para a construção de soluções para a tarefa de REN cria a possibilidade de explorar áreas anteriormente negligenciadas. O domínio jurídico, por exemplo, abundante em dados textuais, é um campo com vasto potencial de exploração. Neste capítulo serão abordados itens que fundamentam a construção deste trabalho.

2.1 Reconhecimento de Entidades Nomeadas

O Reconhecimento de Entidades Nomeadas (REN) consiste em identificar e classificar as entidades existentes em um texto. A classificação das entidades ocorre de acordo com categorias previamente estabelecidas, que variam de acordo com o objetivo da tarefa ou domínio ao qual o texto pertence. Usualmente tem-se categorias como pessoa, organização, local e tempo. A tarefa de REN foi originalmente proposta em 1996 durante a conferência *Sixth Message Understanding Conference* (MUC-6) [Grishman and Sundheim, 1996]. O objetivo do evento era promover e avaliar pesquisas dedicadas a extração de informação. Ao analisar os componentes principais do processo de extração de informação, percebeu-se que uma parte muito importante consistia em identificar e classificar, de forma correta, as entidades nomeadas presentes no texto. O desempenho final da tarefa estava diretamente conectado a esta etapa. Dessa forma, decidiu-se por formalizar a o processo de localizar e iden-

tificar entidades, estabelecendo-se oficialmente a tarefa de REN.

Formalmente, a entrada para a tarefa de REN é uma sequência de T *tokens* $s = \langle w_1, w_2, \ldots, w_T \rangle$. A saída desta tarefa é um conjunto de triplas do tipo (t_s, t_e, c), onde c representa a categoria da entidade e $t_s, t_e \in \{1, 2, \ldots, T\}$ representam, respectivamente, os índices de início e fim de uma entidade. Segue abaixo um exemplo para a sequência:

$$A_{w_1}\ Lei_{w_2}\ Maria_{w_3}\ da_{w_4}\ Penha_{w_5}\ foi_{w_6}\ sancionada_{w_7}\ pelo_{w_8}\ ex\text{-}presidente_{w_9}$$
$$Luiz_{w_{10}}\ Inácio_{w_{11}}\ Lula_{w_{12}}\ da_{w_{13}}\ Silva_{w_{14}}\ em_{w_{15}}\ 7_{w_{16}}\ de_{w_{17}}\ Agosto_{w_{18}}\ de_{w_{19}}\ 2006_{w_{20}}$$

$$\Downarrow$$

Modelo de REN

$$\Downarrow$$

(2, 5, LEGISLAÇÃO)	$\rightarrow$	"Lei Maria da Penha"
(10, 14, PESSOA)	$\rightarrow$	"Luiz Inácio Lula da Silva"
(16, 20, TEMPO)	$\rightarrow$	"7 de Agosto de 2006"

Desta forma, além de classificar as entidades, os modelos de REN também precisam determinar suas delimitações exatas. Isso significa que, para uma entidade composta por mais de uma palavra, é necessário que sejam identificadas todos os componentes da entidade.

O REN, além de uma aplicação por si só, também é importante dentro de outras tarefas de PLN, como Recuperação de Informação, Modelagem de Tópicos e Busca Semântica [Yadav and Bethard, 2019]. Segundo [Guo et al., 2009], cerca de 71% das frases buscadas na internet possuem pelo menos uma entidade nomeada em seu conteúdo. Reconhecer as entidades presentes em strings de busca auxilia a compreensão do contexto da busca e consequentemente pode resultar em melhores resultados.

Para executar a tarefa de REN, é necessária uma coleção de textos, denominada corpus. Essa coleção pode conter textos de estilos diversos, com um caráter generalista, ou pode ser formada apenas por textos de um domínio específico. Neste último caso, a restrição de domínio acarreta algumas particularidades. Características relacionadas a sintática, semântica e morfologia das palavras diferem muito de demais textos. Outro ponto refere-se à classificação das entidades. O tipo de entidade a ser reconhecida pode mudar de acordo com o domínio de trabalho. Enquanto em textos de caráter geral, nomes de pessoas são associados a classe de entidades *PESSOA*, domínios específicos podem tipificar nomes a categorias mais particulares.

Para a construção de um corpus voltado para a tarefa de REN é necessário que se estabeleça um processo de anotação. Esse processo pode ser feito de

maneira automática ou manual. Anotações automáticas são feitas de forma rápida e geram um grande volume de dados anotados. Para realizar as anotações são empregados modelos de REN ou heurísticas. Já os processos de anotação manual são mais demorados, uma vez que são realizados por humanos. Além disso, o número final de documentos anotados também é menor. Contudo, anotações manuais são mais confiáveis e geram resultados mais robustos. Também e necessário definir um esquema de anotação. Os corpora utilizados neste trabalho foram anotados utilizando o esquema IOB [Ramshaw and Marcus, 1995]. Neste tipo de anotação, cada token é associado a um rótulo. Cada rótulo é construído pela junção de um prefixo, seguido da categoria da entidade:

- **B-** indica que o token denota o começo de uma entidade nomeada;
- **I-** indica que o token está contido em uma entidade nomeada;
- **O** indica que o token não pertencem a nenhuma entidade nomeada.

Outro ponto importante relacionado ao REN é a métrica utilizada para avaliação. É importante definir um padrão de avaliação, para que seja possível medir e comparar resultados. Comumente, dois tipos de métricas são encontradas nos trabalhos disponíveis na literatura: métrica por token e métrica por entidade. A métrica por token (ou palavra) leva em conta o número de palavras classificadas de forma correta. Isso significa que, dada uma entidade nomeada formada por três palavras, em um cenário onde todas as entidades foram preditas de forma correta, são contabilizados três acertos. Se apenas uma ou duas palavras foram classificadas conforme a entidade correta, ainda assim se contabiliza um ou dois acertos, respectivamente. Este tipo de métrica potencializa os resultados e por este motivo não é a métrica padrão utilizada. Já a segunda maneira de avaliação consiste na métrica por entidade. Neste caso, o número de acertos contabilizados independe do número de palavras que formam uma entidade nomeada. Se uma entidade nomeada é composta por três palavras e apenas as duas primeiras foram classificadas corretamente, nenhum acerto é contabilizado. Considera-se um acerto quando todos as palavras que compõem uma entidade são classificados de forma correta. Essa é a métrica utilizada em grande parte dos trabalhos de REN, inclusive na realização deste trabalho.

2.2 Domínio Jurídico

É crescente a participação da tecnologia no segmento jurídico. Em relação a inteligência artificial, isso não é diferente. O domínio jurídico apresenta diversas características que o tornam singular e desafiador ao mesmo tempo.

Estilo de escrita particular, diversas categorias de conhecimento (leis, jurisprudências, acórdãos, etc.) e uma vasta coleção de documentos são alguns atributos.

O Poder Judiciário brasileiro gera milhares de documentos todos os anos. Sejam tribunais de instâncias superiores ou comarcas locais, todos os dias são produzidos textos das mais variadas formas. Segundo o Relatório Justiça em Números 2019, divulgado pelo Conselho Nacional de Justiça (CNJ), o Poder Judiciário finalizou o ano de 2018 com 78,7 milhões de processos aguardando uma solução definitiva. Na Figura 2.1 é apresentada a série histórica de casos novos e processos baixados desde o ano de 2009 até 2018. É importante notar que apenas em 2017 o número de processos baixados conseguiu superar o casos novos. Esses números traduzem a crescente demanda por ferramentas que auxiliem o trabalho do judiciário. Um possível alívio na carga de trabalho acumulada pode gerar diversos impactos positivos tanto para os trabalhadores do setor, quanto para a população em geral. A celeridade do processo judiciário pode garantir que o acesso a justiça seja mais universal.

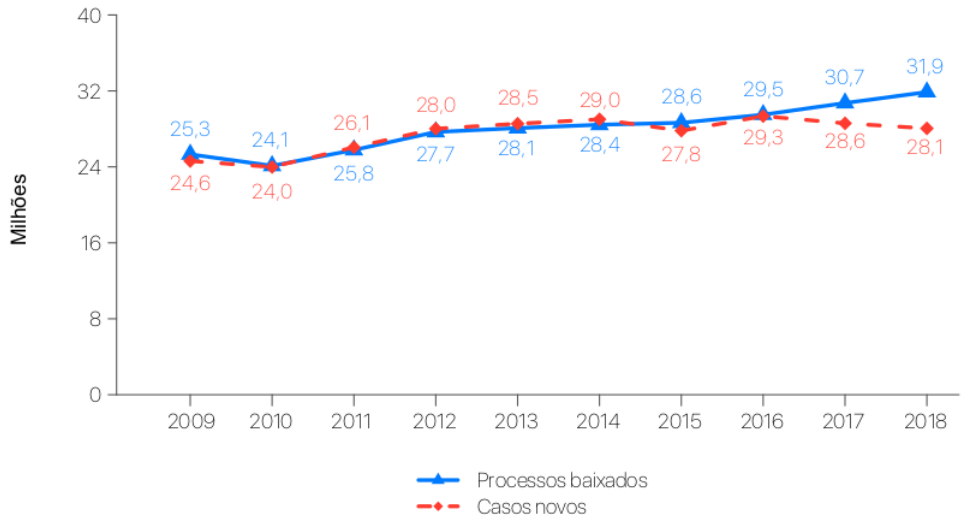


Figura 2.1: Série histórica de casos novos e processos baixados na justiça brasileira. Fonte: Conselho Nacional de Justiça.

Contudo, diversos são os impasses encontrados no momento de integrar a área jurídica à ferramentas de PLN. Algoritmos baseados em Aprendizado de Máquina (AM) ou Aprendizado Profundo (AP) exigem dados anotados para o processo de treinamento. O número de corpora anotados para este domínio é exíguo. Embora seja uma área que disponha de uma quantidade significativa de dados textuais, poucas são as iniciativas para organizar e estruturar textos para construção de conjuntos de dados, anotados ou não. No ano de 2018, em uma iniciativa inédita, [Luz de Araujo et al., 2018] divulgou um corpus construído a partir de documentos legais e com entidades nomeadas manualmente anotadas.

2.3 Modelos Tradicionais de Linguagem

Modelos de linguagem tradicionais abordam a predição da probabilidade de uma palavra w_t considerando a sequência de palavras anteriores $w_{t-1}, w_{t-2}, \ldots, w_1$. Desta forma, conforme descrito por Peters et al. [2017], um modelo de linguagem calcula a probabilidade de uma sequência de palavras $(w_1, w_2, \ldots, w_T)$ como:

$$p(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} p(w_t | w_1, w_2, \ldots, w_{t-1}). \tag{2.1}$$

Por outro lado, alguns modelos profundos de linguagem mais recentes, como BERT e ELMo, utilizam uma modelagem diferente. O modelo ELMo, por exemplo, é bidirecional, pois considera os contextos esquerdo e direito para estimar a distribuição de probabilidade de uma palavra. Já o BERT implementa uma modelagem mascarada de linguagem, que consiste em ocultar palavras durante o treinamento do modelo de linguagem. Além disso, características como a direcionalidade e utilização de word embeddings também apresentam diferenças entre os modelos citados. Os detalhes destes modelos são discutidos nas próximas seções.

2.4 BERT

Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2018] é um codificador de sentenças construído a partir de blocos baseados na arquitetura Transformer [Vaswani et al., 2017]. Destaca-se por utilizar aprendizado profundo, não-supervisionado e bidirecional. A modelagem de linguagem, realizada de forma bidirecional e profunda, é baseada em uma tarefa de pré-treinamento que mascara tokens da sequência de entrada. Utilizando um ML bidirecional, o BERT constrói representações para as palavras considerando os dois sentidos (esquerda para a direita e vice-versa) em uma sequência de entrada. O modelo possui uma arquitetura baseada em camadas. Além das camadas que utilizam codificadores Transformer, o modelo possui uma camada de entrada, dedicada aos word embeddings, camadas formadas por redes *feed-forward*, conexões residuais, módulos de normalização, além de uma camada de saída.

Na Figura 2.2 é ilustrada a arquitetura do BERT que pode ser dividida em três módulos: entrada, atenção e saída. O Módulo de Entrada combina três camadas de embedding para gerar uma representação para cada palavra da sequência de entrada. O Módulo de Atenção contém blocos baseados no codificador Transformer e é responsável pela construção das representações geradas pelo modelo. Neste módulo, os codificadores baseados na arquitetura

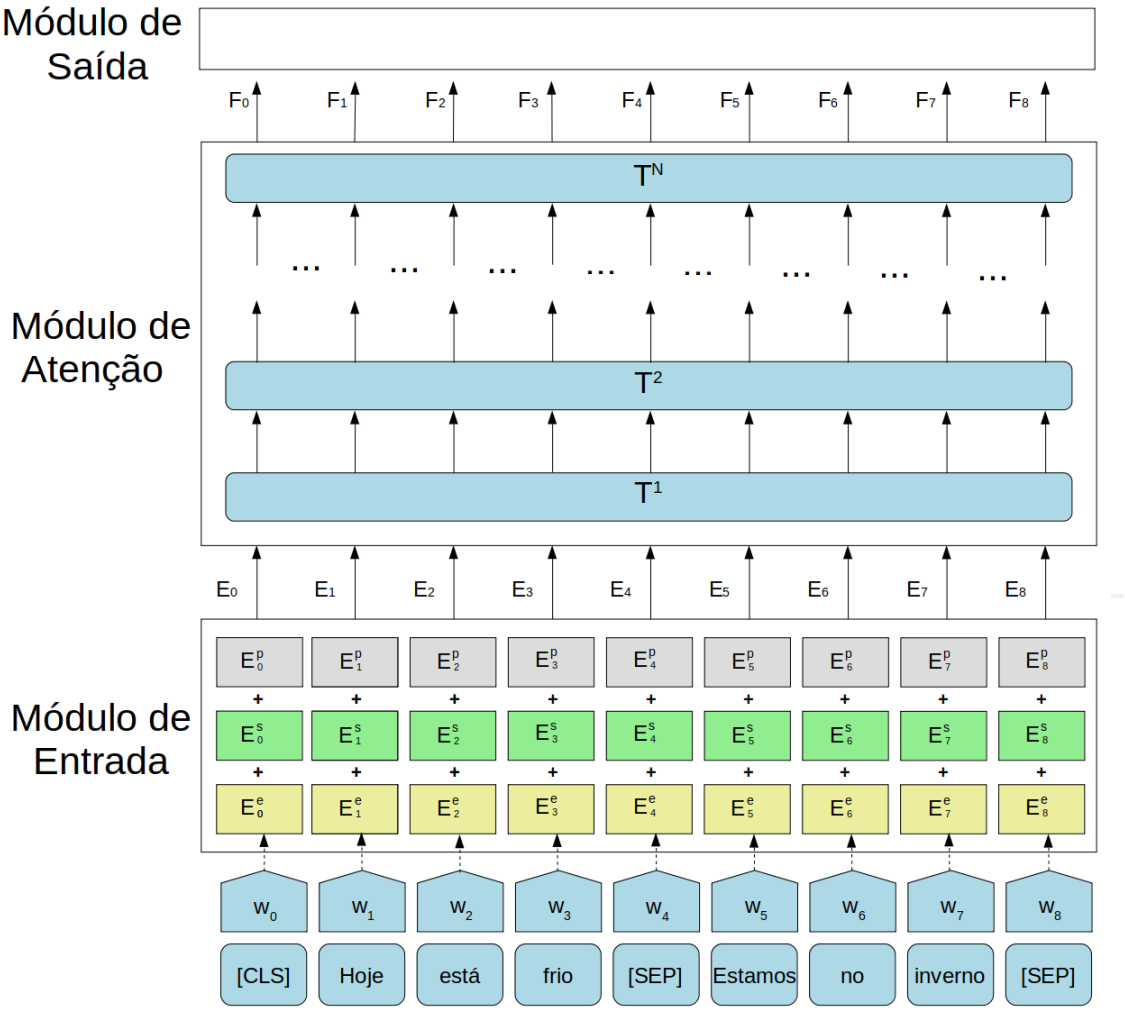


Figura 2.2: Visão geral da arquitetura do modelo BERT organizado em três módulos: Módulo de Entrada, Módulo de Atenção e Módulo de Saída.

Transformer formam diferentes camadas de multi-atenção. Essas camadas são responsáveis por calcular a similaridade entre um determinado token w_t e todos os demais tokens da sequência de entrada. As representações geradas por estas camadas são robustas, uma vez que para cada token representado, todo o contexto da sequência de entrada é considerado. Já o Módulo de Saída apresenta variações de acordo com a tarefa de interesse. A seguir, são discutidos os três módulos desta arquitetura.

2.4.1 Módulo de Entrada

Uma das principais características do BERT diz respeito à vasta gama de tarefas para as quais o modelo pode ser aplicado. O cerne desta versatilidade está na modelagem das sequências de entrada do modelo. Uma sequência de entrada $(w_0, w_1, \ldots, w_T)$ é formada por um segmento de tokens, além dos tokens especiais que marcam o início e o final de uma sentença, conforme

ilustrado na Figura 2.2. O token especial `[CLS]` é adicionado ao início de cada sequência de entrada. Outro token especial, denominado `[SEP]`, é utilizado para separar um par de sentenças em uma sequência[1]. Consequentemente, é possível representar uma única sentença ou um par de sentenças em uma sequência de entrada. O número máximo de tokens em uma sequência deve ser pré-determinado, como um hiperparâmetro do modelo, de acordo com a necessidade de utilização.

A construção do word embedding de cada token é feita com base na abordagem descrita em Wu et al. [2016]. Utilizando o tokenizador WordPiece, cada token pode ser subdividido em um conjunto de sub-tokens. Essa abordagem é capaz de lidar com tokens não pertencentes ao vocabulário de forma vantajosa. Por exemplo, a sentença O cachorro fugiu correndo, ao ser tokenizada pelo WordPiece, resulta em

```
O cachorro fugiu corre ##ndo
```

Sendo que o prefixo ## indica que o sub-token é continuação do sub-token anterior. O verbo correr, conjugado na forma *correndo* foi dividido em dois sub-tokens. Ao tokenizar a sentença *O homem correu na chuva*, temos

```
O homem corre ##u na chuva
```

Comparando estes dois exemplos, é possível perceber que o radical comum às palavras *correndo* e *correu* foi preservado em ambas tokenizações. Em um cenário onde a palavra *correu* não esteja no vocabulário, o modelo não terá visto a palavra durante o treinamento e consequentemente não será capaz de construir uma representação. Ainda assim, caso a palavra apareça durante a validação, sua representação seria próxima a da palavra *correndo*, conforme ilustrado no exemplo.

Além das representações E_i^e de cada token w_i, para $i = 0, 1, ..., T$, o módulo de entrada também considera embeddings de segmento, representados na Figura 2.2 como E_i^s. Para cada token da sequência de entrada, é somado o embedding correspondente ao segmento ao qual aquele token pertence. Esta abordagem está diretamente relacionada com tarefas que utilizam sequências de entrada formadas por duas sentenças. Para os tokens w_i, pertencentes à primeira sentença, é somado o vetor E^0, ou seja, $E_i^s = E^0$. Por outro lado, para os tokens w_i que pertencem à segunda sentença é somado o vetor E^1. Caso seja uma sentença única, o vetor E^0 é somado a todos os tokens. Por fim, para

[1]A terminologia adotada no trabalho original do BERT utiliza a palavra *sentence* para referir-se a um intervalo arbitrário de texto contíguo ao invés de uma frase gramatical. No presente trabalho, seguindo esta notação, o termo *sentença* será usado para este fim. Já a palavra *sequence*, no trabalho do BERT, indica a sequência de tokens de entrada do modelo, que pode ser uma sentença ou um par de sentenças. Aqui, será utilizado o termo *sequência* no mesmo sentido de *sequence* do trabalho original.

cada token w_i são adicionados embeddings E_i^p relacionados à posição de cada token. Como o módulo de atenção não considera a ordem dos tokens em uma sequência, esta abordagem permite que o modelo incorpore informações sobre a posição de cada token. Para cada token da sequência de entrada são somados os embeddings do token, do segmento ao qual ele pertence e da sua posição na sequência. Desta forma, a saída do Módulo de Entrada para um token w_i é dada por:

$$E_i = E_i^e + E_i^s + E_i^p.$$

2.4.2 Módulo de Atenção

O Módulo de Atenção é formado pelo encadeamento de diversas camadas de atenção, conforme pode ser observado na Figura 2.2. Cada camada emprega múltiplas cabeças de atenção (multiatenção), além de conexões residuais e redes *feed-forward*. Cada cabeça consiste em um mecanismo de autoatenção baseado no codificador Transformer [Vaswani et al., 2017]. O conceito de atenção, empregado nestes codificadores, é uma alternativa a convoluções e operações recorrentes.

O módulo de atenção é composto por N camadas de atenção. Como todas estas camadas são iguais (apesar de cada uma ter seu próprio conjunto de parâmetros), iremos descrever aqui uma camada genérica $n \in \{1, 2, \ldots, N\}$, a qual é ilustrada na Figura 2.3. Nesta figura, por simplicidade mas sem perda de generalidade, representamos uma entrada com apenas três tokens. A entrada da camada n é composta pelas representações de saída da camada de atenção anterior para cada token de entrada. Desta forma, a n-ésima camada de atenção toma como entrada a sequência de vetores $(T_0^{n-1}, T_1^{n-1}, \ldots)$ e produz como saída a sequência de vetores $(T_0^n, T_1^n, \ldots)$.

O primeiro passo da camada de atenção é justamente a operação de multi-atenção composta por várias cabeças de atenção. Esta operação, que será detalhada posteriormente, toma a sequência de entrada $(T_0^{n-1}, T_1^{n-1}, \ldots)$ e produz a sequência de saída $(M_0^n, M_1^n, \ldots)$. Em seguida, cada vetor M_i^n, para $i = 0, 2, \ldots$, é somado ao vetor de entrada T_i^{n-1}, representando uma conexão residual. Por fim, o resultado da soma passa por uma camada de normalização produzindo a saída:

$$X_i^n = \mathrm{norm}(T_i^{n-1} + M_i^n). \tag{2.2}$$

A camada seguinte é uma rede *feed-forward*. Esta rede possui duas camadas lineares conectadas por uma função de ativação do tipo ReLU. Sua saída é representada por:

$$Z_i^n = \mathrm{relu}(\mathrm{relu}(X_i^n W_1^n + b_1^n) W_2^n + b_2^n) \tag{2.3}$$

sendo função de ativação ReLU é definida por:

$$\text{relu}(x) = \max(0, x). \tag{2.4}$$

Finalmente, as saídas desta rede são somadas aos vetores X_i^n (conexão residual) e então normalizadas:

$$T_i^n = \text{norm}(X_i^n + Z_i^n). \tag{2.5}$$

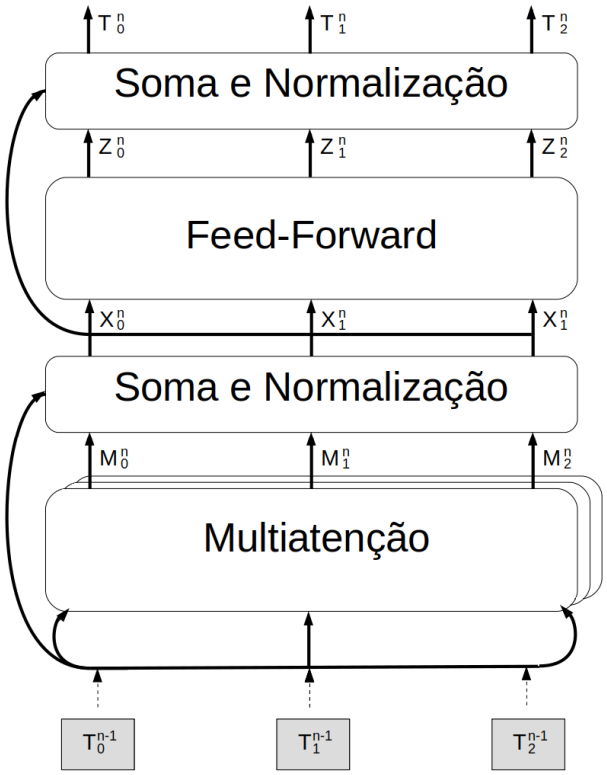


Figura 2.3: Camada $n \in \{1, 2, \ldots, N\}$ de multiatenção do Módulo de Atenção.

A camada de multiatenção representada na Figura 2.3 é formada por C cabeças de atenção. Na Figura 2.4, ilustramos uma camada de multiatenção com duas cabeças. Cada cabeça de atenção é responsável por calcular uma representação diferente para cada token da sequência de entrada. Desta forma, a diversidade das representações aumenta, pois diferentes cabeças possuem diferentes parâmetros e são capazes de "prestar atenção" em diferentes partes da mesma entrada. Todas as cabeças tomam como entrada a sequência $(T_0^{n-1}, T_1^{n-1}, \ldots)$ e cada cabeça $c \in \{1, 2, \ldots, C\}$ fornece como saída uma sequência $(M_0^{n,c}, M_1^{n,c}, \ldots)$ de vetores, um para cada token de entrada. A saída da camada de multiatenção é formada pela concatenação, token a token, das saídas das C cabeças de atenção, Isto é, para o i-ésimo token na n-ésima

camada do Módulo de Atenção, a saída é:

$$M_i^n = [M_i^{n,1}; M_i^{n,2}; \ldots; M_i^{n,C}], \tag{2.6}$$

onde $[\,\cdot\,;\,\cdot\,]$ é o operador de concatenação.

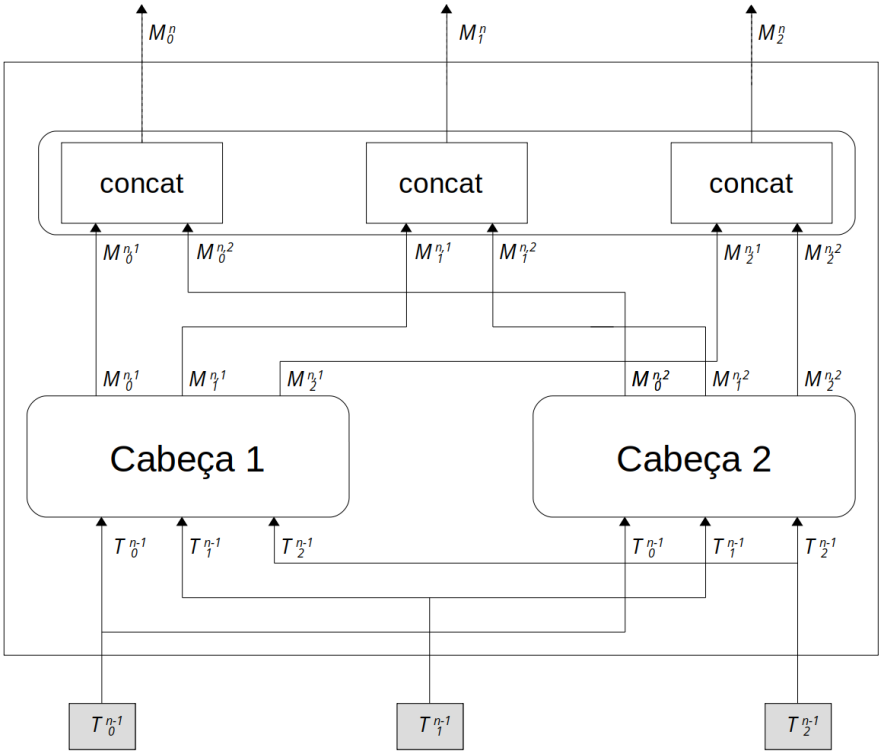


Figura 2.4: Representação da n-ésima camada de multiatenção com duas cabeças de atenção.

Como o funcionamento de todas as cabeças do mecanismo de atenção é idêntico, basta descrever uma cabeça delas. Na Figura 2.5, é representada uma cabeça de atenção c de uma camada n do Módulo de Atenção. Para cada token de uma sequência de entrada, cada cabeça de atenção calcula a semelhança deste token com todos os demais tokens da mesma sequência. Antes deste cálculo, são calculados os vetores Q_i, K_i, e V_i, para o i-ésimo token de entrada. Tais vetores são calculados para cada token de entrada e são denominados índice (Q), chave (K) e valor (V). Os vetores índice e chave são utilizados para calcular a similaridade entre cada par de token. Os vetores valor, juntamente com as medidas de similaridade, são usados para determinar o vetor de saída da cabeça de atenção para cada token. Cada um destes três vetores é obtido por meio de projeções lineares. Para as três projeções são utilizados três matrizes de parâmetros W_{cn}^q, W_{cn}^k e W_{cn}^v para cada cabeça c e cada camada n. Entretanto, por simplicidade, os índices c e n de cada matriz foram omitidos na figura e nas equações correspondentes.

Para tornar a Figura 2.5 mais simples e didática, foi utilizada uma entrada

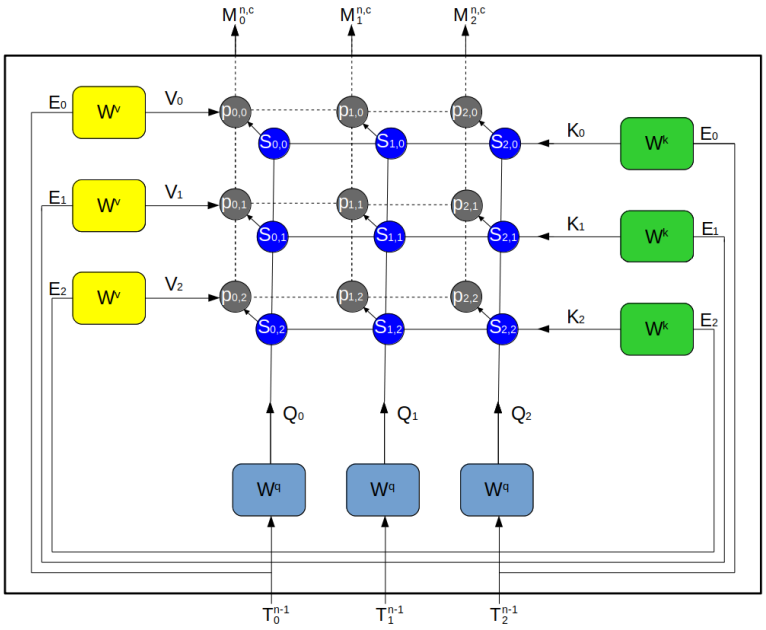


Figura 2.5: Representação do cálculo de atenção, para três tokens de entrada, realizado pela c-ésima cabeça de atenção na n-ésima camada do Módulo de Atenção.

composta por apenas três tokens. As projeções lineares Q_i, K_i, e V_i são definidas por meio de:

$$\begin{aligned} Q_i &= T_i^{n-1} W^q \\ K_i &= T_i^{n-1} W^k \\ V_i &= T_i^{n-1} W^v \end{aligned} \tag{2.7}$$

onde W^q, W^k e W^v são as matrizes responsáveis pelas projeções lineares de cada elemento e T_i^{n-1} é a representação de entrada do i-ésimo token. A similaridade entre um par de tokens (i, j) é definida pelo produto escalar dos vetores Q_i e K_j com uma normalização:

$$S_{i,j} = \frac{Q_i K_j}{\sqrt{d_k}} \tag{2.8}$$

onde d_k representa a dimensão dos vetores Q_i e K_j. Esta normalização controla a variância dos valores de similaridade, evitando problemas de *vanishing gradient* [Vaswani et al., 2017].

Na Figura 2.5, o resultado do cálculo de similaridade entre as representações T_0^{n-1}, T_1^{n-1} e T_2^{n-1} está representado nos círculos de cor azul. Assim, são criadas representações que indicam como cada token de uma sequência se relaciona com os demais. As similaridades $S_{i,0}, S_{i,1}, \ldots$, entre o i-ésimo token e cada um dos outros tokens, são normalizadas por meio de uma operação de softmax:

$$p_{i,j} = \frac{\exp(S_{i,j})}{\sum_{k=0}^{n} \exp(S_{i,k})}. \tag{2.9}$$

Os valores $p_{i,j}$ são ilustrados na Figura 2.5 como círculos de cor cinza. O vetor $p_{i,\cdot}$ contém a similaridade normalizada do i-ésimo token com cada token da sequência de entrada. Por fim, os vetores de valor V_j são ponderados pelos valores $p_{\cdot,j}$ e somados:

$$M_i^{n,c} = \sum_{j=0}^{n} p_{i,j} V_j, \tag{2.10}$$

para se obter a representação $M_i^{n,c}$ do i-ésimo token.

Originalmente, foram propostos dois modelos, com tamanhos distintos, baseados na arquitetura BERT. O tamanho dos modelos está relacionado ao número de módulos de atenção, camadas de multiatenção e dimensão das representações geradas. Consequentemente, essa diferença é refletida no número de parâmetros de cada modelo. O modelo $\text{BERT}_{\text{BASE}}$ utiliza 12 módulos de atenção, enquanto o modelo $\text{BERT}_{\text{LARGE}}$ usa 24. Para cada modelo, os módulos utilizados possuem 12 e 16 camadas de multiatenção, respectivamente. A dimensão das representações geradas também é diferente, sendo de 768 para o primeiro e 1024 para ao segundo. O modelo $\text{BERT}_{\text{BASE}}$ tem 110 milhões de parâmetros treináveis, comparado a 340 milhões para o $\text{BERT}_{\text{LARGE}}$.

2.4.3 Módulo de Saída: Modelo de Linguagem

Para que seja possível treinar o modelo de linguagem bidirecional, o BERT introduz uma tarefa não-supervisionada de pré-treinamento: modelagem de linguagem mascarada. Esta tarefa contorna problemas oriundos da modelagem bidirecional. Adicionalmente, uma segunda tarefa de pré-treinamento também é proposta, denominada predição de próxima sentença. O objetivo desta tarefa é aprender a relação que existe entre duas sentenças.

Modelo de Linguagem Mascarado

A modelagem bidirecional de linguagem percorre uma sentença em dois sentidos, da direita para a esquerda e vice-versa. Quando realizada de forma profunda, as informações recebidas por cada sentido são combinadas durante o processamento. Contudo, a bidirecionalidade do ML introduz um problema. Uma vez que os dois sentidos de uma sentença são percorridos ao mesmo tempo, o ML indiretamente acessa todas as palavras da sequência de entrada, tanto as anteriores como as posteriores. Desta forma, o ML seria capaz de fazer predições de forma trivial, uma vez que todo o contexto está disponível. Para contornar este problema, mantendo a bidirecionalidade, os autores do BERT propõem que uma parcela das palavras seja mascarada. A Modelagem de Linguagem Mascarada (MLM) também é encontrada na literatura com a denominação de Cloze [Taylor, 1953]. A ideia se resume a substituir aleatori-

amente alguns tokens por uma máscara, representada por um token especial `[MASK]`. Um exemplo é apresentado a seguir.

```
[CLS] Suspensas as sessões no Superior [MASK] Federal [SEP]
```

No trabalho original do BERT é proposto que, durante o pré-treinamento do ML, 15% das palavras sejam mascaradas. Durante o treinamento, o modelo deve predizer quais palavras foram mascaradas. Embora essa alternativa permita a existência da bidirecionalidade, ela também gera outro efeito indesejado. Durante o treinamento do ML, serão encontrados diversos tokens `[MASK]`. Após isso, quando o modelo for submetido a um processo de *finetuning* para uma tarefa específica, tokens `[MASK]` nunca ocorrerão. Se o modelo fosse treinado apenas para predizer os tokens `[MASK]` e então não encontrasse esse token durante o *finetuning*, ele estabeleceria que não há necessidade de predizer nada. A única representação contextual que seria aprendida pelo ML seria aquela relacionada ao token mascarado, resultando em um aprendizado enviesado. Para evitar que o ML atente-se apenas aos tokens removidos, faz-se então a seguinte distribuição dentre os 15% de tokens mascarados:

- 80% dos tokens são substituídos por `[MASK]`,
- 10% dos tokens são substituídos por uma palavra aleatória, e
- 10% dos tokens não são alterados.

É importante ressaltar que essa substituição, mesmo que seja simples, deve ser feita com cautela. Substituir palavras de forma aleatória em uma sentença pode introduzir ruído, causando confusão no modelo e deteriorando os resultados. Por este motivo, a substituição por palavras aleatórias é feita em apenas 10% dos 15% de tokens selecionados. Isso corresponde a apenas 1.5% de todos os tokens utilizados para o treinamento do ML. A função de perda (loss function) do BERT leva em consideração apenas a predição dos valores mascarados e ignora a predição dos demais. Uma consequência direta disso é que o modelo leva mais tempo para convergir se comparado a modelos unidirecionais.

Predição de Próxima Sentença

Utilizando a tarefa de MLM, o modelo consegue capturar informações das palavras e de como elas se relacionam em uma sentença. Ainda é necessário que o ML seja capaz de absorver as relações existentes entre duas sentenças distintas, quais suas dependências de contexto e como estão conectadas. Para isso, é proposto um mecanismo binário de Predição de Próxima Sentença (PPS). No momento em que as sentenças são escolhidas para criar

uma sequência de entrada, conforme descrito na Seção 2.4.1, dois tipos de sequência são criados. No primeiro tipo, duas sentenças *subsequentes* A e B são escolhidas do corpus de treinamento. Desta forma, a sentença B é a continuação do que está descrito na sentença A, conforme o exemplo a seguir:

```
[CLS] O homem foi [MASK] mercado [SEP]
Ele comprou uma garrafa [MASK] água [SEP]
```

Já o segundo tipo de sequência consiste em duas sentenças A e B onde B é uma sentença aleatória retirada do corpus. Ou seja, geralmente, B não possui nenhuma relação com A, como demonstrado no exemplo:

```
[CLS] O homem foi [MASK] mercado [SEP]
Pinguins são aves [MASK] não voam [SEP]
```

A distribuição entre os dois tipos de sequência ocorre numa proporção igual, ou seja, 50% para cada tipo. Desta forma, o modelo aprende a reconhecer se duas sentenças de um mesmo segmento estão relacionadas contextualmente ou não.

2.4.4 Modelo de Saída: REN

Para que seja possível aplicar o modelo anteriormente descrito à tarefa de REN, é necessário adicionar uma camada de classificação a saída do modelo, substituindo o Módulo de Saída, representado na Figura 2.2. A camada de classificação de tokens é acoplada à saída do Módulo de Atenção. Desta forma, as representações geradas pelo BERT são utilizadas por essa camada de classificação. Na Figura 2.6 é ilustrado como o modelo utiliza as representações para realizar a tarefa final. Como a tarefa de REN pode ser definida

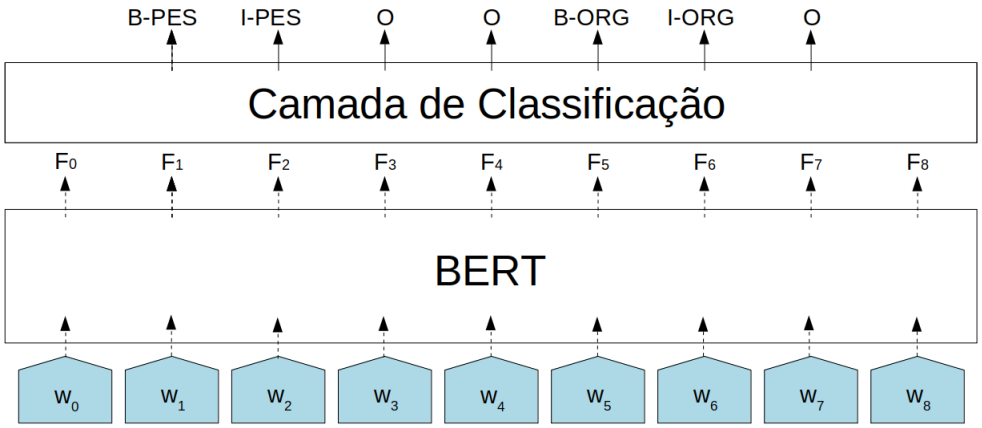


Figura 2.6: Representação da arquitetura do modelo aplicado a tarefa de REN. Uma camada de classificação é utilizada para classificar as saídas geradas pelo BERT.

como uma tarefa de classificação de tokens, a camada de classificação é responsável por atribuir um rótulo para cada token da sequência de entrada. Na Figura 2.6 são apresentados alguns exemplos de classificação, onde as representações são classificadas com os rótulos `B-PES` e `I-PES`, por exemplo. Essa categoria indica que os dois tokens correspondem ao nome de uma pessoa. Além disso, a classificação também inclui rótulos do tipo `O`. Apesar de não representar nenhuma entidade nomeada, é importante que o modelo seja capaz de identificar corretamente este tipo de token. É importante destacar que os tokens `[CLS]` e `[SEP]` não são utilizados na etapa de classificação. Durante o treinamento supervisionado para a tarefa, o modelo ajusta os pesos de todas as camadas, além dos pesos da camada de classificação. Contudo, essa etapa consome menos recursos se comparada com a fase de pré-treinamento. As tarefas de pré-treinamento, principalmente a tarefa de modelagem de linguagem mascarada, requer grande quantidade de texto. Além disso, durante o pré-treinamento, o modelo realiza uma distribuição de probabilidades a partir de todos os tokens do vocabulário. Já no treinamento supervisionado para REN, como não são realizadas as tarefas de modelagem de linguagem e o número de classes a serem preditas é menor, o tempo envolvido é menor.

2.5 ELMo

ELMo (Embeddings from Language Models) [Peters et al., 2018] são representações de palavras construídas de forma contextual com a capacidade de modelar características complexas das palavras. As representações são construídas por meio dos estados internos de um modelo de linguagem bidirecional e pré-treinado. Durante o pré-treinamento do modelo de linguagem, a representação de uma palavra é calculada por uma função que considera o contexto global da sequência a qual a palavra pertence, uma vez que o significado de uma palavra é totalmente dependente do contexto em que ela está inserida De acordo com os autores, essa modelagem é capaz de capturar relações sintáticas e semânticas, além de identificar ocorrências de polissemia. Como se trata de um modelo bidirecional, ele também percorre a sequência de entrada em duas direções. Apesar disso, a combinação dessas informações é feita de forma independente. Os detalhes deste modelo serão discutidos a seguir.

2.5.1 Arquitetura do Modelo

A arquitetura do modelo pode ser dividida três camadas: camada de entrada (ou camada de embedding), camada de modelagem de linguagem e camada de saída. Na Figura 2.7 é ilustrada a arquitetura do modelo. De maneira geral, a camada de entrada é responsável pela construção dos word embed-

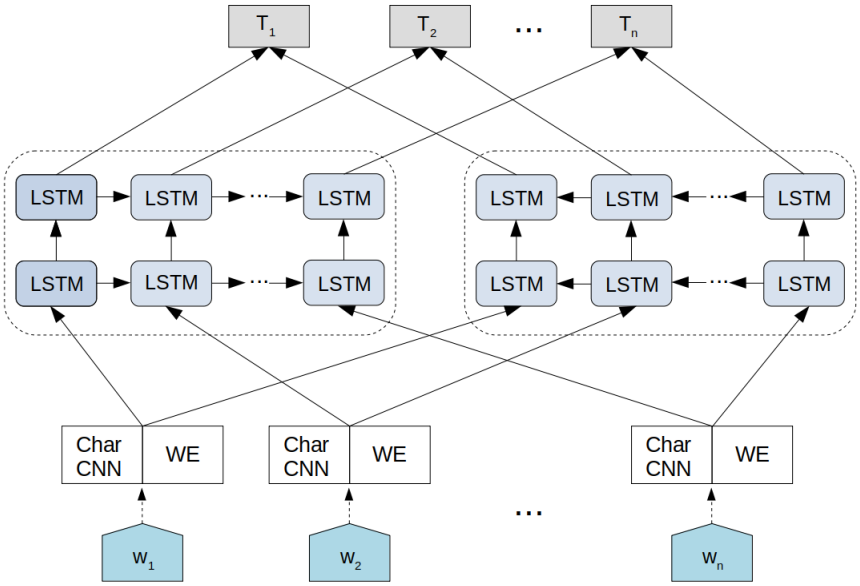


Figura 2.7: Representação da arquitetura do ELMo.

dings utilizados pelo modelo. Já a camada de representações implementa a modelagem de linguagem proposta. Por fim, na camada de saída são obtidas as predições para as palavras. A seguir serão discutidos os principais conceitos relacionados a camada de entrada e a camada de modelagem de linguagem. Como as representações de interesse são obtidas pelos estados intermediários do modelo de linguagem, a camada de saída não será discutida.

2.5.2 Camada de Entrada

Embora o objetivo do ML seja fornecer representações para as palavras, faz-se necessário fornecer algum tipo de representação inicial para a entrada do modelo. Dessa forma, vetores estáticos pré-treinados são utilizados para a construção da representação inicial desta camada. Além destes vetores, outra representação, relacionada aos caracteres que formam cada token, também é utilizada. Essas duas representações são combinadas, formando o word embedding resultante da camada de entrada.

A representação relacionada aos caracteres que formam cada token, são obtidas por uma rede convolucional, identificada na Figura 2.7 por *CharCNN*. Essa rede possui filtros de convolução com tamanhos variáveis, além de uma camada de *max-pooling* junto a saída. A representação a nível de caracteres permite que o modelo utilize características morfológicas relacionadas as palavras, criando representações robustas mesmo para palavras que não pertençam ao dicionário. Essa representação é concatenada com as representações estáticas, indicadas por *WE*, conforme ilustrado na Figura 2.7. Cada *WE* representa um vetor estático pré-treinado de word embeddings. As repre-

sentações resultantes desta concatenação são utilizadas como entrada para a camada de modelagem de linguagem.

2.5.3 Modelagem de Linguagem

A camada de modelagem de linguagem é responsável por construir as representações ELMo. Conforme anteriormente mencionado, o ELMo também emprega modelagem de linguagem bidirecional. Denominado biLM (bidirectional Language Model), o modelo utilizado pelo ELMo é construído por dois modelos de linguagem que funcionam de forma unidirecional. Isto significa que cada modelo atua em uma direção distinta, um percorre o texto da esquerda para a direita e o outro no sentido oposto. Conforme apresentado na Figura 2.7, a camada de entrada do modelo recebe uma sequência de entrada $(w_i, w_2, \ldots, w_n)$. As representações geradas pela camada de entrada são utilizadas pela camada que processa a entrada no sentido de leitura do texto, ou seja, da esquerda para a direita. O mesmo acontece para a camada que realiza o processamento no sentido inverso. Seja uma sequência de n tokens representada por $(w_1, w_2, ..., w_n)$, o ML que percorre a sequência da esquerda para a direita calcula a distribuição dessa sequência por meio da modelagem da probabilidade do token w_k em $(w_1, \ldots, w_{k-1})$:

$$p(w_1, w_2, \ldots, w_N) = \prod_{k=1}^{N} p(w_k | w_1, w_2, \ldots, w_{k-1}) \tag{2.11}$$

De modo similar, um segundo ML é responsável por percorrer a sequência da direita para a esquerda de forma a predizer o token w_k dado contexto futuro:

$$p(w_1, w_2, \ldots, w_N) = \prod_{k=1}^{N} p(w_k | w_{k+1}, w_{k+2}, \ldots, w_N) \tag{2.12}$$

Apesar de ser bidirecional, o biLM combina as representações geradas por cada ML, que atuam de maneira independente. As representações ELMo são uma combinação dos estados intermediários do biLM. Para cada token w_k de uma sequência de entrada, é gerado um conjunto de $2L + 1$ representações, onde L representa o número de camadas do biLM.

$$R_k = \{\mathbf{x}_{\mathrm{k}}^{\mathrm{LM}}, \overrightarrow{\mathbf{h}}_{\mathrm{kj}}^{\mathrm{LM}}, \overleftarrow{\mathbf{h}}_{\mathrm{kj}}^{\mathrm{LM}} | j = 1, ..., L\} = \{\mathbf{h}_{\mathrm{kj}}^{\mathrm{LM}} | j = 0, ..., L\} \tag{2.13}$$

em que $\mathbf{x}_{\mathrm{k}}^{\mathrm{LM}}$ é a representação criada pela convolução de caracteres para cada token, $\overrightarrow{\mathbf{h}}_{\mathrm{kj}}^{\mathrm{LM}}$ e $\overleftarrow{\mathbf{h}}_{\mathrm{kj}}^{\mathrm{LM}}$ são as representações calculadas por cada biLSTM, relacionadas a sentidos distintos. A representação ELMo de um token w_k é um vetor obtido a partir da representação R_k. Na abordagem mais simples, o vetor

ELMo é obtido pelas representações geradas pelas camadas superiores de R_k. Para a tarefa de REN, são utilizadas as representações geradas por todas as camadas, não apenas a última. Porém, o processo de treinamento também envolve um parâmetro relacionado à tarefa específica

$$\mathbf{ELMo}_{\mathrm{k}}^{\mathrm{tarefa}} = \gamma^{\mathrm{tarefa}} \sum_{j=0}^{L} s_{\mathrm{j}}^{\mathrm{tarefa}} \mathbf{h}_{\mathrm{kj}}^{\mathrm{LM}} \tag{2.14}$$

Os pesos s^{tarefa} são aprendidos para cada tarefa específica. Já o parâmetro escalar γ^{tarefa} é utilizado para corrigir possíveis distorções entre as distribuições resultantes do biLM e as distribuições relacionadas à tarefa específica.

CAPÍTULO

3

Metodologia Experimental

Os modelos profundos de linguagem discutidos anteriormente foram utilizados para as avaliações experimentais deste trabalho. Por meio do processo de *finetuning*, foram derivados novos modelos de diferentes naturezas. Cada modelo resultante foi avaliado em tarefas de REN. Os corpora utilizados, bem como as especificações das etapas de *finetuning* e treinamento supervisionado serão apresentados nesta seção.

3.1 Visão Geral

A metodologia aplicada neste trabalho é focada em uma avaliação experimental organizada em quatro componentes. Estes componentes são ilustrados na Figura 3.1 e listados abaixo:

(i) modelo profundo de linguagem pré-treinado em domínio geral;

(ii) *finetuning* não supervisionado de (i) em domínio específico;

(iii) treinamento supervisionado do modelo para a tarefa de REN; e

(iv) avaliação do modelo para REN utilizando corpus de validação.

O componente (i) consiste em MLs pré-treinados em dados de domínio geral. Os MLs adotados para a avaliação experimental, descritos no Capítulo 2, são baseados nas arquiteturas BERT e ELMo. Foram selecionados MLs pré-treinados nos corpora Wikipédia e brWaC, ambos de domínio geral. É importante destacar que o ML baseado na arquitetura BERT foi pré-treinado de maneira multi-idioma, considerando fontes de idiomas diferentes.

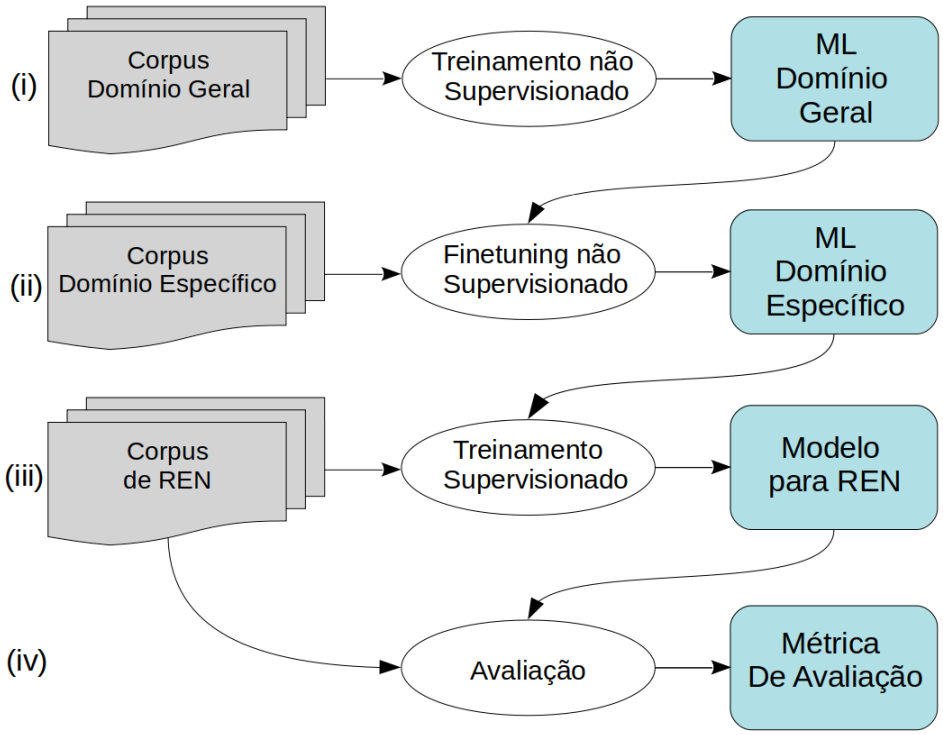


Figura 3.1: Diagrama da metodologia composta por quatro componentes.

O componente (ii) refere-se à etapa de *finetuning* não supervisionado do modelo resultante de (i). Utilizando o corpus Acórdãos-TCU, de domínio jurídico e sem anotações, cada ML genérico foi submetido ao processo de *finetuning*, derivando assim MLs específicos para o domínio jurídico. Conforme indicado na Figura 3.1, os modelos resultantes deste componente são utilizados para o treinamento supervisionado, descrito pelo componente (iii).

O componente (iii) representa a etapa de treinamento supervisionado na tarefa de REN, enquanto o componente (iv) corresponde à fase de validação. Nesta fase, os modelos resultantes de (ii) são utilizados para inicializar o treinamento supervisionado para a tarefa de REN, por meio de um corpus anotado. Finalmente, o componente (iv) corresponde à fase de avaliação dos modelos resultantes de (iii). Nesta etapa, os modelos são submetidos a validações na tarefa de REN. Os componentes (iii) e (iv) apresentam um elevado grau de dependência. Isso acontece porque o corpus utilizado para o treinamento supervisionado, descrito no componente (iii), e o corpus utilizado na etapa de avaliação, referente ao componente (iv), são o mesmo. Embora o corpus seja o mesmo, conjuntos distintos são utilizados por cada componente.

Além das etapas descritas pelos componentes anteriormente elencados, também foram realizadas avaliações com o propósito de estabelecer resultados de base. Para estas avaliações, os MLs de domínio geral relacionados ao componente (i) foram submetidos diretamente aos componentes (iii) e (iv). Desta forma, obteve-se avaliações destes modelos nas tarefa de REN. Estes resulta-

dos são utilizados para validar o impacto da proposta deste trabalho, uma vez que exprimem o desempenho dos modelos sem o processo de *finetuning* em dados de domínio específico.

Com relação aos componentes (iii) e (iv), além de corpora de domínio específico para a tarefa de REN, também foi utilizado o corpus HAREM, de domínio geral, para o treinamento supervisionado e posterior avaliação. Por meio da avaliação dos modelos neste corpus, deseja-se verificar o efeito do *finetuning* intradomínio em MLs, quando avaliado em corpus de domínio geral.

3.2 Modelos de Linguagem de Domínio Geral

Neste trabalho, consideram-se MLs baseados em duas arquiteturas: BERT e ELMo. Para cada uma das arquiteturas escolhidas, tem-se disponíveis versões pré-treinadas dos seus respectivos MLs. Essas versões foram pré-treinadas em enormes quantidades de texto de domínio geral sem qualquer tipo de anotação.

O ML baseado na arquitetura BERT foi pré-treinado em textos de diversos idiomas. O corpus utilizado para o pré-treinamento englobava os 100 idiomas com as maiores Wikipédias. Este modelo pré-treinado foi disponibilizado oficialmente pela equipe do Google no repositório oficial do projeto[1]. Aqui este modelo será referenciado como BERT Multilingual. Para a implementação deste trabalho, utilizou-se a biblioteca `transformers` da Huggingface[2]. Diferente da versão original disponibilizada pelo Google, essa versão está implementada com base no framework PyTorch.

Para o modelo baseado na arquitetura ELMo, foram utilizados dois MLs pré-treinados, ambos utilizando dados de domínio geral. O pré-treinamento de ambos MLs utilizou apenas textos no idioma Português, caracterizando uma versão mono-idioma. O primeiro modelo foi pré-treinado utilizando a Wikipédia em Português. Esse modelo será referenciado como ELMo Wikipédia. O segundo ML foi pré-treinado no corpus brWaC, disponibilizado pelos trabalhos de Castro et al. [2018] e de Castro [2019]. Aqui, este ML será mencionado como ELMo brWaC. Estes dois MLs estão disponíveis na seção de contribuições da página oficial do ELMo[3].

Embora seja possível pré-treinar MLs do início, ou seja, inicializar os modelos com pesos aleatórios e atualizá-los durante o treinamento, esse tipo de abordagem apresenta enorme complexidade computacional. Particularmente para a arquitetura BERT, o treinamento do ML com a tarefa de palavras mascaradas requer enorme poder computacional [Devlin et al., 2018]. Além disso,

[1] https://github.com/google-research/bert
[2] https://github.com/huggingface/transformers
[3] https://allennlp.org/elmo

com a disponibilidade de modelos pré-treinados em enormes quantidades de dados, é mais interessante o *finetuning* destes modelos, derivando-se novos MLs de acordo com os objetivos específicos da tarefa a ser resolvida.

3.2.1 Corpora Não Anotados de Domínio Geral

Com a evolução dos modelos de aprendizagem profunda, os corpora sem anotações tornaram-se imprescindíveis. Como não são necessárias anotações, estes corpora são formados por enormes coleções de textos. Geralmente, os textos são obtidos de alguma fonte suficientemente grande, como a Wikipédia, por exemplo. Entretanto, existem iniciativas que buscam compilar em um único corpus textos obtidos a partir de diversas fontes. O corpus brWaC [Wagner Filho et al., 2018], por exemplo, é um corpus construído a partir de textos coletados de inúmeras páginas da internet.

Na Tabela 3.1, são exibidas estatísticas básicas relacionadas aos corpora sem anotações utilizados neste trabalho. A seguir, estes corpora serão detalhados.

Corpus	Documentos	Frases	Palavras
Wikipédia	934k	13,900k	1,400mi
brWaC	3,530k	145,000k	2,680mi
Wikipédia Multilingual	100 maiores Wikipédias		

Tabela 3.1: Avaliação quantitativa dos corpora não anotados de domínio geral.

Wikipédia

O corpus Wikipédia foi criado a partir da coleta de todos os artigos da Wikipédia que pertencentes ao Português. Após a coleta dos artigos, foram removidos conteúdos como hiperlinks, citações e referências e marcações HTML. Utilizando a biblioteca spaCy[4], também foram removidos itens de pontuação em duplicidade e artigos sem qualquer conteúdo. Por último, cada artigo foi adicionado ao arquivo com corpus, com uma linha em branco separando artigos diferentes.

brWaC

O corpus brWaC [Wagner Filho et al., 2018] é constituído pelo texto de páginas coletadas da internet. Mais de 60 milhões de endereços foram utilizados para construção do corpus, todos pertencentes ao domínio brasileiro (*.br*). De acordo com os autores, a construção do corpus foi realizada em três

[4]https://spacy.io/

etapas. Na primeira etapa, foram identificados os conjuntos de URLs. Um motor de buscas foi utilizado para encontrar conteúdos. As entradas do motor de busca foram geradas de forma aleatória. Os dez primeiros resultados para cada busca foram coletados. Ao final dessa etapa, foram obtidas mais de 38 milhões de URLs. O segundo passo concentra-se em filtrar os documentos anteriormente coletados, estabelecendo uma série de critérios. Levou-se em conta o tamanho do texto, marcações HMTL e densidade de *stop words*. Este último parâmetro serviu como indicador para assegurar que os documentos selecionados realmente pertenciam ao idioma Português. A etapa final dedicou-se à remoção de conteúdos duplicados ou documentos com algum tipo de intersecção.

Wikipédia Multilingual

Este corpus, utilizado durante o pré-treinamento do modelo BERT Multilingual, é formado pela agregação de Wikipédias em diferentes idiomas. Mais precisamente, foram coletados todos os artigos das cem maiores Wikipédias considerando diferentes idiomas. O número de documentos que formam o corpus é de aproximadamente 50 milhões, com um vocabulário de 119 mil tokens.

3.3 Modelos de Linguagem de Domínio Específico

Conforme ilustrado na Figura 3.1, o Componente (ii) da metodologia consiste no *finetuning* não supervisionado dos MLs de domínio geral em um corpus de domínio específico. Nesta etapa, os modelos utilizados como base foram o BERT Multilingual e o ELMo brWaC. O corpus de domínio específico utilizado para o *finetuning* foi o Acórdãos-TCU. Por esta razão, os modelos resultantes serão referenciados como BERT Acórdãos e ELMo Acórdãos, respectivamente. Adicionalmente, derivou-se um outro modelo por meio do *finetuning* do BERT Multilingual utilizando o corpus Wikipédia. O objetivo disto é estudar o impacto do processo de *finetuning* de um modelo multi-idioma utilizando textos apenas em Português que é a língua da tarefa alvo. Este modelo será referenciado como BERT Wikipédia.

3.3.1 Corpus Não Anotado de Domínio Específico

Ainda que seja mais fácil coletar textos de diversas fontes de domínio geral, existem iniciativas voltadas à construção de corpora não-anotados construídos apenas por textos de domínios específicos, O corpus Acórdãos-TCU é um exemplo. Este conjunto é formado apenas por documentos jurídicos, todos

oriundos da mesma fonte. Na Tabela 3.2 são exibidas as estatísticas básicas do corpus. Construído a partir da coleta de dados do Tribunal de Contas da

Acórdãos-TCU	
Documentos	298k
Frases	9,000k
Palavras	912mi

Tabela 3.2: Avaliação quantitativa do corpus Acórdãos-TCU, formado apenas por documentos jurídicos.

União, o corpus Acórdãos-TCU[5] é composto por acórdãos proferidos pelo tribunal entre 1992 e 2019, coletados por *web crawling* e disponibilizados sem qualquer tipo de tratamento adicional. Para ser aplicado a este trabalho, a única etapa de processamento necessária consistiu em separar as frases de cada parágrafo, com o intuito de fragmentar cada frase em uma linha específica do corpus.

3.4 Treinamento Supervisionado de REN

A transferência de um modelo de linguagem profundo para realizar uma tarefa final consiste em conectar uma camada (ou sequência de camadas) à camada superior do modelo de linguagem. Este processo está representado pelo componente (iii) na Figura 3.1. Geralmente, para realizar a tarefa de REN, são utilizadas as representações geradas por MLs como as entradas de um modelo de classificação de tokens. Dessa forma, quando o modelo construído para a tarefa final é treinado de forma supervisionada, os pesos do ML podem ser atualizados ou não, de acordo com a abordagem adotada.

Após concluída a etapa de *finetuning* do MLs, todos os modelos resultantes são submetidos ao treinamento supervisionado para a tarefa de REN. Para este treinamento, foram utilizados os corpora HAREM (domínio geral), LeNER-Br (domínio jurídico) e DrugSeizures-Br (domínio jurídico). O corpus HAREM, mesmo sendo de domínio geral, foi utilizado nas etapas de treinamento supervisionado e avaliação. Desta forma, é possível distinguir se possíveis ganhos de desempenho dos MLs de domínio específico são causados pelo *finetuning* em dados intradomínio ou somente pelo aumento na quantidade de dados de treinamento. A seguir serão discutidas as metodologias adotadas para o treinamento supervisionado em relação a cada uma das arquiteturas utilizadas neste trabalho.

[5]https://github.com/netoferraz/acordaos-tcu

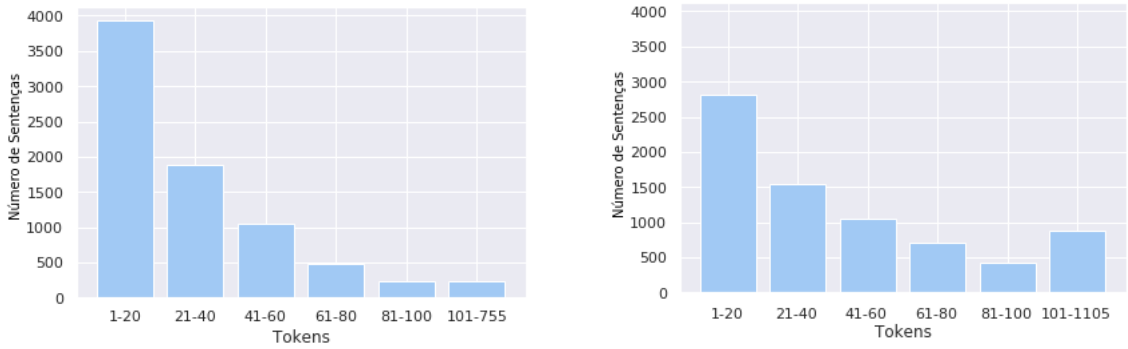
3.4.1 BERT

Uma característica marcante em modelos baseados na arquitetura BERT é a necessidade de poucas alterações arquiteturais para aplicação em diferentes tarefas. Para utilizar as representações geradas pelo modelo para a tarefa de REN, adiciona-se uma camada de classificação diretamente conectada à saída do modelo de linguagem. Estas representações são geradas pelo módulo de atenção, conforme descrito na Seção 2.4.2, formadas por um vetor para cada token da sequência de entrada. As representações são utilizadas pela camada de classificação de forma direta. De acordo com a Figura 2.2, estas representações são denominadas $(F_1, F_2, \ldots, F_n)$, onde n é o número máximo de tokens da sequência de entrada.

A camada de classificação utilizada é responsável por realizar uma projeção a partir das representações resultantes do ML para um espaço vetorial reduzido, que considera apenas a quantidade de rótulos utilizada para a classificação das entidades nomeadas. Esta projeção é definida como:

$$y_i = \text{softmax}(WF_i + b), \tag{3.1}$$

onde W representa os parâmetros da camada de classificação, b é o valor de bias e $i \in \{1, 2, \ldots, n\}$ representa o índice de um token da sequência de entrada. O resultado consiste na distribuição de probabilidades dos rótulos da tarefa de REN para o i-ésimo token da sequência de entrada. Durante o treinamento supervisionado, além de treinar a camada de classificação, também são ajustados os pesos calculados durante o pré-treinamento não supervisionado. Isso permite que o modelo se ajuste à distribuição do conjunto de dados utilizado para o treinamento supervisionado. Durante o treinamento supervisionado na tarefa final, também é utilizado o processo de tokenização descrito na etapa de treinamento não supervisionado do BERT. Neste processo, uma palavra pode ser dividida em sub-palavras, o que pode resultar em um aumento do número de tokens por frase, após concluído o processo. Quando observado, este comportamento pode acarretar alguns problemas, especialmente em corpora do domínio jurídico. Geralmente, textos do domínio jurídico são formados por frases longas. Na Figura 3.2, são exibidas as distribuições do número de tokens por frase no conjunto de treinamento do LeNER-Br. Enquanto em (a) é exibida a distribuição antes do processo de tokenização do BERT, em (b) é apresentado o número de tokens após o processo de tokenização. Vale notar que antes da tokenização, a maior frase do conjunto de teste tinha 755 tokens. Contudo, esse número sobe para 1.105 tokens após a aplicação do processo. Um dos parâmetros do BERT está diretamente relacionado às informações acima mencionadas. O modelo obedece a uma *quantidade máxima*



(a) Antes da tokenização.

(b) Após a tokenização.

Figura 3.2: Distribuição do número de tokens nas frases do conjunto de treinamento do LeNER-Br antes e depois da tokenização.

de tokens em uma sequência. O maior valor permitido para esse atributo é de 512. Segundo Devlin et al. [2018], sequências muito longas devem ser evitadas, uma vez que o mecanismo de atenção tem complexidade de execução quadrática em relação ao tamanho da sequência. Entretanto, se um número muito pequeno fosse definido para esse parâmetro, o modelo poderia ignorar diversas entidades nomeadas distribuídas no corpus. As sentenças com mais tokens do que esse número máximo seriam truncadas, não sendo possível saber quantas entidades seriam descartadas. Ademais, descartar entidades afetaria diretamente o processo de validação dos modelos. Uma vez que entidades poderiam ser descartadas, o número total de entidades consideradas no processo de validação e teste estaria comprometido.

Como este problema pode ser observado em textos de diversos domínios, no trabalho original do BERT é descrita uma abordagem relacionada ao contexto do documento, quando o modelo é aplicado à tarefa de REN. Contudo, não são fornecidos muitos detalhes sobre esta abordagem. No trabalho de Souza et al. [2019], uma abordagem baseada em janela de deslocamento de contexto é utilizada. Sentenças onde a quantidade de tokens é maior do que o valor máximo permitido são divididas em sentenças menores. Essa divisão é feita por meio de uma janela de tamanho fixo, que percorre a sentença a ser dividida. Neste trabalho, não foram utilizadas as abordagens anteriormente mencionadas. Ainda que os conjuntos de treinamento e validação possuam sentenças longas, as avaliações inicias mostraram que nenhuma entidade foi descartada, mesmo considerando o truncamento destas sentenças.

Os MLs de domínio geral BERT Multilingual e BERT Wikipédia e o ML de domínio específico BERT Acórdãos foram submetidos ao treinamento supervisionado nos seguintes corpora: HAREM, LeNER-Br e DrugSeizures-Br. O modelo BERT Multilingual determina os resultados utilizados como base de comparação. Para cada uma das tarefas supervisionadas consideradas, os hiperparâmetros

de treinamento foram ajustados com base no conjunto de desenvolvimento correspondente.

3.4.2 ELMo

Para que fosse possível aplicar os modelos baseados no ELMo a tarefa de REN, utilizou-se a biblioteca AllenNLP[6]. A implementação fornecida por esta biblioteca implementa a modelagem de linguagem do ELMo, descrita na Seção 2.5.3. Para que seja possível aplicar o modelo à tarefa de REN, é utilizada uma uma camada de classificação, denominada CRF (Conditional Random Fields) [Lafferty et al., 2001], responsável pela classificação dos tokens. O CRF é um modelo discriminativo muito utilizado em tarefas de classificação de tokens. Utilizando as classificações para cada token de uma sequência de entrada, ele calcula valores de pontuação associados a cada transição entre as possíveis classificações observadas.

Seja uma sequência de entrada definida como $X = (x_1, x_2, \ldots, x_n)$, define-se M como a matriz formada pelas representações do modelo de linguagem, cujas dimensões são $n \times k$, onde n é o número de tokens da sequência de entrada e k representa o número de classes a serem preditas. Dessa forma, $M_{i,j}$ corresponde ao valor de pontuação da classe j para o token i da sequência X. De acordo com Lample et al. [2016] para uma sequência de predições $y = (y_1, y_2, \ldots, y_n)$ as pontuações são calculadas por

$$s(X, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=0}^{n} P_{i, y_i} \tag{3.2}$$

em que A é a matriz de transições, onde $A_{i,j}$ marca a pontuação de transição da classe i para a classe j e P representa a matriz de representações, gerada pelo ML. Durante o treinamento, busca-se maximizar as probabilidades da sequência classificada de forma correta por meio de

$$log(p(y|X)) = s(X, y) - log(\sum_{y' \in Yx} e^{s(X, y')}) \tag{3.3}$$

onde Yx representa todas as sequências possíveis para a sequência de entrada x.

Foram utilizados os word embeddings do tipo GloVe, pré-treinados [Hartmann et al., 2017] em Português. Este material foi disponibilizado pelo NILC[7] (Núcleo Interinstitucional de Linguística Computacional). Os word embeddings utilizados possuem dimensão de tamanho 300.

Durante o treinamento supervisionado, os vetores ELMo de um token são

[6] https://allennlp.org./
[7] http://www.nilc.icmc.usp.br/embeddings

calculados por meio da combinação dos estados internos gerados pelo ML. Para o treinamento supervisionado foram utilizados os modelos ELMo Wikipédia, ELMo brWaC e ELMo Acórdãos. O modelo ELMo Wikipédia determina os resultados utilizados como base de comparação. Seguindo a metodologia adotada para os modelos baseados na arquitetura BERT, o ajuste dos parâmetros de treinamento também foram realizados utilizando o conjunto de desenvolvimento. Entretanto, o treinamento supervisionado considerou apenas os seguintes corpora: LeNER-Br e HAREM. Os corpora anotados utilizados para o treinamento supervisionado e validação dos modelos de REN são apresentados a seguir.

3.4.3 Corpora Anotados de REN

Para resolver tarefas em PLN utilizando técnicas de aprendizagem de máquina, torna-se indispensável a utilização de corpora. Desde uma coleção de textos minimamente estruturada até conjuntos manualmente anotados, eles têm um papel elementar na construção de soluções. Embora considerada esta importância, diversos idiomas, incluindo o português, sofrem com a escassez de recursos. Não são tantas iniciativas para a criação de corpora quando comparamos com o inglês, por exemplo. A situação é ainda pior quando considera-se domínios específicos. Grande parte dos corpora disponíveis são de propósito geral. Contudo, algumas iniciativas para promover a construção e avaliação de corpora em português obtiveram sucesso. É o caso do HAREM, uma competição organizada com o intuito de promover a tarefa de REN em Português. Além de chamar atenção para a difusão da tarefa, os corpora construídos durante o evento tornaram-se o objeto de avaliação de muitos trabalhos disponíveis na literatura.

Recentemente, foi proposto um corpus formado apenas por documentos jurídicos brasileiros. O corpus LeNER-Br foi construído a partir de anotações manuais. Além disso, como contribuição deste trabalho, é apresentado o corpus DrugSeizures-Br. Na Tabela 3.3 é apresentada a composição dos corpora anteriormente discutidos.

Corpus	Documentos	Frases	Palavras
HAREM	257	8k	156k
LeNER-Br	70	10k	318k
DrugSeizures-Br	6,218	118k	6,400k

Tabela 3.3: Comparação entre o número de documentos, frases e palavras para os corpora anotados.

3.4.4 HAREM

Organizado pela Linguateca[8], o HAREM é um evento para avaliação de modelos de REN para o Português. As duas edições do evento resultaram os seguintes conjuntos: HAREM I [Santos and Cardoso, 2007] e HAREM II [Mota and Santos, 2008]. Além disso, um evento intermediário apresentou o MiniHAREM [Cardoso, 2006]. Esses corpora são utilizados em diversos trabalhos [Castro et al., 2018, Souza et al., 2019, dos Santos and Guimarães, 2015, dos Santos Neto, 2019] que avaliam modelos de REN para o Português. Na Tabela 3.4 são exibidas estatísticas básicas sobre os conjuntos do HAREM utilizados neste trabalho. Por simplicidade, o termo HAREM será utilizado para identificar os dois conjuntos utilizados neste trabalho: HAREM I e MiniHAREM, conforme a divisão proposta em dos Santos and Guimarães [2015]. O HAREM I foi dividido entre treino e validação. O conjunto de validação corresponde a 10% do conjunto de treino. Já o conjunto de teste é formado integralmente pelo MiniHAREM. Ao total, o HAREM possui dez entidades anotadas. Na Tabela 3.5 exibi-se o número de entidades por conjunto do HAREM no cenário total.

Conjunto	Documentos	Frases	Palavras
HAREM I	129	4749	156k
MiniHAREM	128	3393	318k
Total	257	8142	474k

Tabela 3.4: Dados relacionados aos conjuntos do HAREM utilizados neste trabalho.

Categoria	Treino	Validação	Teste
Pessoa	1,010	20	832
Tempo	404	32	361
Local	1,129	108	877
Organização	885	40	625
Valor	422	41	326
Obra	189	7	190
Coisa	128	7	170
Acontecimento	125	3	57
Abstração	372	34	228
Outro	37	3	28
Total	4,701	295	3,694

Tabela 3.5: Estatística básica para os conjuntos do HAREM.

[8]https://www.linguateca.pt/

Em muitos casos, é possível encontrar trabalhos que dividem o HAREM em dois cenários: total e seletivo. Enquanto o cenário total considera as dez entidades presentes na anotação original, o cenário seletivo possui apenas cinco entidades. Neste último cenário, as entidades originais são agrupadas em cinco super-tipos: Pessoa, Organização, Local, Valor e Data. Na Tabela 3.6 são apresentados os conjuntos de entidades nomeadas para o cenário seletivo,

Categoria	Treino	Validação	Teste
Pessoa	1,010	20	832
Tempo	404	32	361
Local	1,129	108	877
Organização	885	40	625
Valor	422	41	326
Total	3,850	241	3,021

Tabela 3.6: Estatística básica para os conjuntos do HAREM no cenário seletivo.

3.4.5 LeNER-Br

Considerado o primeiro corpus de REN em português dedicado ao domínio jurídico, o LeNER-Br [Luz de Araujo et al., 2018] é formado apenas por documentos jurídicos. Para a sua construção, coletou-se 66 documentos de fontes como Supremo Tribunal Federal, Superior Tribunal de Justiça e Tribunal de Contas da União. Além disso, outros 4 documentos relacionados à legislação brasileira também foram adicionados ao corpus. O LeNER-Br possui pouco mais do que 300 mil tokens, assim como os corpora Paramopama [Júnior et al., 2015] e CONLL-2003 [Tjong Kim Sang and De Meulder, 2003]. O corpus foi dividido em três conjuntos: treino, validação e teste. De forma prática, em cada arquivo dos conjuntos anteriormente citados, há uma palavra por linha seguida pelo rótulo referente à anotação da entidade nomeada daquela palavra. Como neste tipo de tarefa é importante identificar o início e fim das frases, uma linha em branco delimita o fim de uma frase e o início de outra.

O processo de anotação das entidades nomeadas foi conduzido de forma manual. Foram anotadas entidades tradicionalmente encontradas em outros corpora como *Pessoa, Organização, Local* e *Tempo*. Adicionalmente, por se tratar de um corpus de domínio específico, duas entidades de cunho jurídico também foram anotadas: *Legislação e Jurisprudência*. A entidade *Legislação* indica citações no texto que estejam relacionadas a leis brasileiras. Já a entidade *Jurisprudência* está associada a decisões anteriores proferidas por

tribunais de justiça. O esquema de anotação adotado foi o IOB [Ramshaw and Marcus, 1995]. Na Tabela 3.7 é apresentado um extrato do conjunto de validação.

A	O	Tribunal	O
Secretaria	B-ORGANIZAÇÃO	,	O
de	I-ORGANIZAÇÃO	por	O
Controle	I-ORGANIZAÇÃO	intermédio	O
Externo	I-ORGANIZAÇÃO	da	O
no	O	Decisão	B-JURISPRUDENCIA
Estado	B-LOCAL	1.040/2002	I-JURISPRUDENCIA
do	I-LOCAL	–	O
Rio	I-LOCAL	Plenário	B-ORGANIZACAO
de	I-LOCAL	,	O
Janeiro	I-LOCAL	determinou	O
-	O	a	O
Secex/RJ	B-ORGANIZAÇÃO	abertura	O

Tabela 3.7: Fragmento retirado do conjunto de validação. Cada linha é formada por uma palavra, um espaço em branco e a anotação da entidade nomeada correspondente.

De acordo com os autores, para a construção dos conjuntos, foi adotada a seguinte distribuição de documentos: 50 documentos para o conjunto de treino, 10 documentos para o conjunto de validação e 10 documentos para o conjunto de teste. O número de entidades por conjunto é apresentado na Tabela 3.8.

Categoria	Treino	Validação	Teste
Pessoa	1,525	310	233
Tempo	1,334	234	192
Organização	2,400	561	501
Local	611	109	47
Legislação	1,920	397	378
Jurisprudência	1,104	207	185
Total	8,894	1,818	1,536

Tabela 3.8: Número de entidades nomeadas por conjunto no LeNER-Br.

3.4.6 DrugSeizures-Br

DrugSeizures-Br é um dataset contendo 6,218 petições relacionadas a procedimentos de apreensão de drogas. Estas petições foram produzidas pelo Ministério Público do Estado de Mato Grosso do Sul (MPMS) entre os anos de 2015 e 2019. O dataset inclui uma tabela de metadados relacionados

às petições que compreendem entidades de interesse do MPMS. As entidades anotadas estão divididas em 25 tipos diferentes envolvendo, por exemplo, nome e quantidade da droga apreendida, dados da pessoa denunciada (nome, documentos, etc.), local da apreensão, dentre outros. Entretanto, o esquema de anotação deste dataset difere das abordagens tradicionais de REN. Uma anotação em um corpus de REN inclui a posição exata onde a menção à entidade ocorre no texto. Por outro lado, as anotações de entidades no dataset DrugSeizures-Br estão relacionadas apenas à denúncia como um todo. Não existe nenhuma referência à posição no texto da denúncia onde a entidade foi mencionada. Para alinhar as anotações ao texto, foi aplicado um algoritmo simples de busca de string. A partir de uma entidade atrelada a uma denúncia, o algoritmo busca a string da entidade no texto da denúncia. Foi utilizado um algoritmo de busca aproximada. Mais especificamente, o algoritmo de busca é baseado na distância de edição entre duas strings[9]. Uma anotação de entidade é criada quando a string da entidade é encontrada no texto da denúncia, respeitando uma distância de edição menor do que 25% do tamanho da string (considerando substituições, adicões e remoções de caracteres). As estatísticas das entidades anotadas desta maneira são apresentados na Tabela 3.9.

Por fim, o corpus resultante foi dividido em treino, validação e teste na proporção de 70%, 10% e 20%, respectivamente. Inspirados pela divisão adotada no HAREM (veja Seção 3.4.4) foram agrupados as 25 categorias de entidades nomeadas em 6 super-tipos: Local, Pessoa, Tempo, Organização, Droga e Outro. A partir deste agrupamento, um corpus com cenário seletivo foi criado considerando cinco entidades nomeadas (ignorando o super-tipo Outro). Na Tabela 3.10 é exibida a distribuição das entidades no cenário seletivo.

Todos os corpora de REN são anotados utilizando o esquema IOB descrito na Seção 2.1. Este esquema é baseado em classificação de tokens e considera três rótulos para cada tipo de entidade nomeada. Esses rótulos indicam que determinado token é o começo de uma entidade (B de *begin*), está dentro de uma entidade (I de *inside*) ou não representa nenhuma entidade (O de *outside*).

3.5 Validação dos Modelos de REN

Conforme descrito na Seção 3.4, os modelos derivados pelo processo de *finetuning* foram treinados de maneira supervisionada em corpora dedicados a tarefa de REN. A etapa de avaliação destes modelos é feita utilizando os mesmos corpora do treinamento supervisionado. Todavia, os conjuntos utilizados para cada etapa são distintos. Para a avaliação dos modelos, são utilizados

[9] https://pypi.org/project/fuzzysearch/

os conjuntos de desenvolvimento ou teste. Avaliações realizadas por meio do conjunto de desenvolvimento tem como principal objetivo verificar a efetividade da etapa de treinamento supervisionado. Desta forma, é possível avaliar os resultados com o intuito de alterar os parâmetros de treinamento, para que seja possível repeti-lo. Após realizadas as alterações e finalizada a etapa de treinamento supervisionado, avaliam-se os modelos resultantes no conjunto de teste de cada corpora. Essa avaliação acontece uma única vez, para que não seja adicionado nenhum viés a avaliação.

Categoria *Super-Tipo*/Tipo	Treino	Validação	Teste
Local			
Nome da Rua	5,386	807	1,556
Número da Rua	2,799	424	789
Bairro	4,762	769	1,426
Cidade	18,412	2,737	5,477
Complemento	202	51	97
Cidade de Origem	3,927	529	1,130
Estado de Origem	5,066	670	1,324
Cidade de Destino	3,764	460	1,030
Estado de Destino	7,283	966	2,118
Pessoa			
Indiciado	26,304	3,351	7,671
Testemunha Ativa	8,977	945	1,601
Testemunha	4,346	654	862
Testemunha Passiva	443	64	86
Investigado	1,242	201	362
Vítima	1,180	173	433
Autor	3,162	446	834
Réu	1,186	122	218
Advogado	67	1	3
Tempo			
Horário	746	95	227
Data	715	145	222
Organização			
Promotoria	140	16	31
Autoridade	3,042	490	815
Droga			
Nome	13,711	2,089	3,862
Outro			
Tipo Penal	5,938	870	1,825
Quantidade da Droga	3,128	403	848
Total	125,928	17,478	34,847

Tabela 3.9: Quantidade de entidades nomeadas por tipo em cada conjunto do corpus DrugSeizures-Br.

Categoria	Treino	Validação	Teste
Pessoa	46,907	5,957	12,070
Tempo	1,461	240	449
Local	51,601	7,413	14,947
Organização	3,182	506	846
Droga	13,711	2,089	3,862
Total	116,862	16,205	32,174

Tabela 3.10: Quantidade de entidades nomeadas para as cinco categorias do cenário seletivo do DrugSeizures-Br.

CAPÍTULO

4

Trabalhos Relacionados

Desde que a tarefa de REN foi definida, diversos tipos de soluções foram propostas. Inicialmente, os modelos eram baseados em regras ou ontologias construídas de forma manual. Surgiram então modelos baseados em extração de características, construídos por meio de aprendizado de máquina. Quando os modelos baseados em aprendizado profundo foram propostos para diversas tarefas de PLN, incluindo a tarefa de REN, as demais abordagens perderam espaço.

Nesta seção, são discutidos os principais trabalhos dedicados a tarefa de REN, incluindo trabalhos direcionados ao Português, bem como trabalhos comprometidos com domínios específicos.

4.1 Reconhecimento de Entidades Nomeadas

As propostas de soluções para a tarefa de REN, baseadas em abordagens tradicionais, exigiam a confecção de atributos ou características produzidas de forma manual. Estes itens eram utilizados como elementos fundamentais no processo de treinamento de modelos. Modelos baseados em regras ou ontologias são um clássico exemplo deste tipo de abordagem. As regras tentam modelar características relacionadas a sintática [Cucchiarelli and Velardi, 2001], semântica [Han and Zhao, 2009], ortografia ou mesmo informações relacionadas ao domínio das entidades. Abordagens baseadas em dicionários [Muñoz et al., 2016], tentavam estabelecer uma definição prévia de todas as entidades presentes em um texto ou domínio. Os modelos baseados nesta abordagem, assim como aqueles baseados em regras, sofriam drasticamente com a falta de capacidade de generalização. Utilizadas como fonte principal de informa-

ção, as regras ou dicionários precisavam cobrir integralmente todos os aspectos das entidades nomeadas para que fosse possível identifica-las completamente. Contudo, na prática esses modelos não conseguiam muito sucesso. O processo de se antecipar a todas as entidades e seus possíveis comportamentos em um texto não é trivial. Além disso, como as regras eram específicas para cada conjunto de dados, isso impossibilitava que as abordagens fossem aproveitadas para diferentes domínios. Outra abordagem clássica aplicada a tarefa de REN utilizava sistemas baseados em características (*feature-based*) [Zhou and Su, 2002]. Nesta abordagem, as características presentes nos textos bem como nas entidades nomeadas eram previamente modeladas, ou seja, deseja-se representar as características de forma a evidencia-las. Para todos os cenários, a construção de atributos ou caraterísticas de forma manual precisava ser feita por especialistas do domínio de interesse da tarefa final, consumindo muitos recursos e tempo. Por estes motivos, com o surgimento dos modelos baseados em aprendizado profundo, este tipo de abordagem perdeu espaço.

Pioneiro em utilizar redes neurais para a tarefa de REN Collobert et al. [2011] propôs o uso de uma arquitetura baseada em redes neurais convolucionais. Sem qualquer tipo de pré-processamento, a rede recebe como entrada as frases de um texto e treina suas camadas para extrair as características relevantes presentes na entrada. Os benefícios resultantes dessa abordagem são fatores muito importantes quando considerado o sucesso dessas redes. Além de dispensar o uso de qualquer característica manualmente construída, a quantidade de dados utilizados para o treinamento também é menor. Além disso, as redes são treinadas de forma fim-a-fim, ou seja, o aprendizado ocorre integralmente em relação aos dados utilizados. Muitos foram os trabalhos que propuseram melhorias na arquitetura anteriormente citada [Huang et al., 2015, Lample et al., 2016, Chiu and Nichols, 2015]. Entretanto, a natureza das redes convolucionais tornava-se um problema quanto ao mapeamento de longas dependências. Em muitos textos ou mesmo frases, as ideias expressadas pelas palavras são retomadas por de referências ao longo do texto. Algumas vezes essas referências estão próximas, mas em muitos casos, estão distribuídas pelo texto. Por isso, ser capaz de identificar tais dependências se torna importante. Neste contexto, as redes recorrentes apresentaram-se como uma alternativa. Este tipo de rede constrói suas representações levando em conta todos os estados anteriores vistos pela rede. Isso significa que, ao processar as palavras de uma sequencia de entrada, a rede considera as palavras que apareceram para a construção da representação da palavra atual. As redes LSTM [Hochreiter and Schmidhuber, 1997] são exemplos de redes recorrentes. Elas são construídas com mecanismos que permitem que longas

dependências sejam armazenadas por longos períodos, por uma célula de memória. Diversos trabalhos voltados para a tarefa de NER utilizam este tipo de arquitetura. Em seu trabalho, Huang et al. [2015] utiliza LSTM juntamente com CRF. Os trabalhos de Ma and Hovy [2016], Chiu and Nichols [2015] empregam arquiteturas semelhantes. As redes LSTM foram largamente utilizadas para soluções de diversas tarefas em PLN. Recentemente, novas arquiteturas com abordagens que dispensam convoluções ou recorrências mostraram-se muito poderosas, o que resultou numa mudança de paradigma.

Considerando trabalhos dedicados a tarefa de REN no idioma Português, Pirovani and Oliveira [2018] apresentam um modelo baseado em um classificador do tipo CRF, em conjunto com regras de gramática local, considerando uma abordagem tradicional. Essas regras são construídas de acordo com as entidades a serem reconhecidas. Utilizando um modelo baseado em LSTM-CRF Castro et al. [2018] avalia a utilização de diferentes word embeddings (FastText, Wang2Vec e Word2Vec) para a tarefa de REN. Embora a utilização de diferentes word embeddings pré-treinados possa apresentar melhorias de desempenho na tarefa de REN, realizar o pré-treinamento para cada método de construção dessas representações é um processo custoso. Neste trabalho, a utilização de diferentes MLs para a construção das representações das palavras dispensa a etapa de pré-treinamento, uma vez que versões pré-treinadas estão disponíveis. Com os resultados obtidos, o trabalho propõe uma combinação de parâmetros que melhor se encaixa para a tarefa de REN no Português.

Para explorar diferentes tipos de arquitetura voltadas para aprendizagem profunda [Fernandes et al., 2018] aplica quatro modelos distintos para a tarefa de REN em Português. Após a realização de seus experimentos, os autores concluem que apesar de existir uma queda nos valores mensurados se comparados a avaliações em corpora do idioma inglês, ela pode ser atribuída a diversos fatores. Contudo, o idioma e a arquitetura dos modelos não podem ser considerados como parte desses fatores que prejudicam o desempenho final.

4.2 Finetuning Intradomínio

Apesar do sucesso alcançado por diversos modelos baseados em aprendizado profundo, muitos domínios apresentavam um limitante comum: escassez de dados anotados. Embora os modelos propostos fossem capazes de extrair características dos dados de treinamento, a quantidade de dados anotados necessários para treinamento supervisionado ainda era muito grande. Fazia-se necessária alguma iniciativa capaz de tirar proveito da enorme quantidade

de dados sem anotações, com a finalidade de mitigar a quantidade necessária de anotações para treinamento supervisionado. Em uma proposta que considerava um passo intermediário de pré-treinamento, Dai and Le [2015] apresentou a seguinte conclusão: o uso de dados não anotados durante o pré-treinamento aumenta a capacidade de generalização do modelo. Seus experimentos mostraram que ao aumentar a quantidade de dados utilizados na tarefa de pré-treinamento, os resultados obtidos na tarefa final eram explicitamente melhores. A etapa de pré-treinamento utilizada consistia em predizer a próxima palavra em uma sentença, utilizando as palavras anteriores. Utilizando a arquitetura proposta em Sutskever et al. [2014], os autores utilizam um codificador para a etapa de pré-treinamento. Após treinado nos dados sem anotações, os parâmetros resultantes são utilizados para inicializar a camada de classificação.

Com o sucesso das abordagens baseadas em transferência de aprendizagem, a proposta de modelos pré-treinados em corpora formados por grandes quantidades de texto sem anotações constitui um marco na área de PLN. Tais modelos atingiram resultados impressionantes em diversas tarefas, mostrando-se tão versáteis quanto poderosos. Ainda assim, estes modelos não repetiam o mesmo sucesso quando utilizados para resolver tarefas de domínios específicos. Nestes casos, apenas o pré-treinamento em dados de domínio geral não é suficiente para que os modelos consigam se adaptar de maneira satisfatória as particularidades presentes em textos de domínio específico. Para contornar este problema, Howard and Ruder [2018] propôs realizar *finetuning* de modelos de linguagem antes de aplicá-los a uma tarefa específica. O trabalho batizado de ULMFiT (Universal Language Model Fine-tuning) descreve um método capaz de aplicar transferência de aprendizado para tarefas de PLN. Utilizando um ML pré-treinado, é realizada uma etapa de *finetuning* utilizando o corpus da tarefa de interesse, antes de realizar o treinamento supervisionado. O *finetuning* tem como objetivo adaptar o ML para as característica existentes no domínio específico da tarefa final. Independente da distribuição do corpus genérico, na maioria das vezes o domínio da tarefa final apresenta comportamento linguístico distinto. Dessa forma, o ML passa por um ajuste em relação as características do domínio alvo.

Baseado na abordagem utilizada pelo ULMFiT, Rother and Rettberg [2018] realizaram pré-treinamento de um ML baseado, utilizando a Wikipédia alemã como fonte de dados. Com o interesse de realizar a tarefa de identificação de discurso de ódios em tweets, os autores realizaram o *finetuning* do ML em dados intradomínio. O modelo submetido ao *finetuning* obteve perplexidade com cerca de três pontos de diferença se comparado ao ML inicial. Chakrabarty et al. [2019] também realiza *finetuning* de um ML baseado no ULMFiT, voltado

para a identificação de reivindicações em argumentos.

A técnica de *finetuning* de LMs ainda é pouco explorada na literatura. Neste trabalho, a técnica de *finetuning* de MLs é explorada considerando diferentes domínios para o pré-treinamento e posterior *finetuning*. Assim como proposto em Howard and Ruder [2018], e conforme descrito no Capítulo 3 o processo de *finetuning* de MLs é realizado em dados intradomínio, relacionados ao domínio da tarefa de interesse. Contudo, o corpus utilizado neste processo não diretamente relacionado com a tarefa final, apesar de compartilharem o mesmo domínio. Recentemente, Gururangan et al. [2020] apresentou um estudo comparativo envolvendo quatro domínios distintos e oito tarefas de classificação. As duas abordagens propostas consideram o *finetuning* de MLs em dois aspectos: utilizando dados relacionados ao domínio da tarefa final e utilizando dados relacionados a tarefa final, mas não necessariamente do mesmo domínio. Os experimentos conduzidos mostraram que o desempenho dos modelos submetidos ao *finetuning* foi superior em todas as tarefas avaliadas. Desta mesma forma, por meio do *finetuning* em dados relacionados ao domínio da tarefa final, este trabalho avalia o impacto desta etapa utilizando a tarefa de REN, que também envolve corpora de domínios geral e específico.

4.3 Domínio Jurídico

É crescente o número de iniciativas que buscam aplicar soluções baseadas em modelos de inteligência artificial para problemas da área jurídica. É possível encontrar trabalhos relacionados a sumarização de documentos [Galgani et al., 2012], Perguntas e Respostas [Kim and Goebel, 2017] e Extração de Informações [Chalkidis and Androutsopoulos, 2017]. Para a tarefa de REN, Dozier et al. [2010] propõe um modelo capaz de identificar juízes, advogados, jurisdições e tribunais em documentos do domínio jurídico. A construção do modelo é baseada em regras de contexto. Trabalhos recentes mostraram como o uso de MLs contribui com melhores resultados para a tarefa de REN. [de Castro et al., 2019] explora transferência de aprendizado em um modelo com representações construídas a partir de um modelo baseado na arquitetura ELMo. O pré-treinamento do modelo utilizou dois corpora distintos: Wikipédia, constituído de textos de domínio geral, além de textos da Justiça do Trabalho. Após o pré-treinamento do modelo em dados de domínio geral, os autores realizaram o processo de *finetuning* do ML em dados do domínio jurídico. Os resultados obtidos mostram que o processo de *finetuning* tem impacto positivo nos resultados da tarefa de REN. Além disso, os autores disponibilizaram o ML pré-treinado do ELMo[1] para a língua portuguesa. Este modelo

[1] https://allennlp.org/elmo

pré-treinado foi utilizado para a realização deste trabalho, conforme descrito na Seção 3. de Castro [2019] apresenta novos resultados utilizando a mesma arquitetura citada anteriormente. Nesta abordagem, o autor explora diferentes combinações de word embeddings aplicados para a mesma arquitetura, de modo a identificar a combinação que traria o melhor desempenho para a tarefa de REN. Além disso, o autor ainda descreve a construção de um corpus voltado para o reconhecimento de entidades, construído apenas por documentos da Justiça do Trabalho. Outro modelo recentemente proposto [Souza et al., 2019] alcançou novos resultados estado-da-arte no HAREM. Utilizando representações geradas pelo BERT e uma camada para classificação (CRF), os autores exploram abordagens baseadas em extração de características e *finetuning* supervisionado. Recentemente, os autores supracitados também divulgaram uma versão pré-treinada para o ML do BERT. Essa versão é totalmente dedicada ao Português. O corpus utilizado para o pré-treinamento do ML foi o brWaC. Por se tratar de um modelo pré-treinado dedicado ao Português, este modelo também foi adotado para a fase de avaliação experimental deste trabalho.

CAPÍTULO

5

Resultados Experimentais

Os experimentos realizados neste trabalho buscam avaliar o impacto do *finetuning* intradomínio de MLs quando avaliados por meio da tarefa de REN. O processo de *finetuning* é realizado tanto em dados de domínio geral, quanto em dados relacionados ao domínio da tarefa de interesse (intradomínio). Posteriormente, os modelos foram avaliados por meio de tarefas de REN de domínio geral e específico. Neste capítulo, são discutidas as métricas de avaliação utilizadas, os hiperparâmetros utilizados nas etapas de avaliação experimental bem como os modelos de linguagem derivados por meio do processo de *finetuning*. Também são apresentadas as avaliações realizadas e os resultados obtidos, em companhia das discussões pertinentes.

5.1 Métrica de Avaliação

A métrica de avaliação utilizada por este trabalho é a F1-Score. Ela combina os valores de precisão e revocação em um único número. Mais especificamente, F1-Score é a média harmônica entre precisão e revocação. Para a tarefa de REN, todos os tokens de uma sequência recebem uma classificação, mesmo que o token não represente uma entidade nomeada. Embora existam diversas entidades nomeadas em um texto, esse número é majoritariamente menor se comparado ao número de tokens que não representam entidade alguma [LI et al., 2009]. Por este motivo, ao se definir uma métrica de avaliação para a tarefa de REN, é necessário determinar que os valores sejam calculados em relação as entidades nomeadas presentes no texto, e não em relação a cada token. Dessa maneira, para que seja possível obter o valor de F1-Score, calculam-se as métricas de *Precisão* e *Revocação*.

A métrica de precisão indica a proporção de entidades nomeadas que foram corretamente classificadas pelo modelo, dentre todas as entidades preditas pelo modelo. O seu valor é definido como:

$$P = \frac{VP}{VP + FP}, \tag{5.1}$$

onde VP (verdadeiros positivos) é a quantidade de entidades nomeadas classificadas corretamente e FP (falsos positivos) é a quantidade de entidades nomeadas classificadas de forma incorreta. Já a métrica de revocação indica, dentre todas as entidades nomeadas que deveriam ter sido identificadas, qual foi a proporção de entidades as quais a classificação correta realmente foi atribuída. Seu valor é calculado como:

$$R = \frac{VP}{VP + FN}, \tag{5.2}$$

onde FN (falsos negativos) indica a quantidade de entidades nomeadas que deixaram de ser identificadas pelo modelo.

Como a tarefa de REN envolve várias classes de entidades nomeadas, os contadores VP, FP e FN definidos acima podem ser calculados para cada classe. Da mesma forma, podemos calcular precisão e revocação por classe. Por outro lado, é desejável calcular valores médios para as métricas anteriormente discutidas. Para tal, é necessário obter os valores médios de precisão e revocação considerando todas as classes avaliadas. Estes valores, também chamados de $micro\text{-}P$ e $micro\text{-}R$, são definidos como:

$$micro\text{-}P = \frac{\sum\limits_{c} VP_c}{\sum\limits_{c} VP_c + FP_c} \tag{5.3}$$

$$micro\text{-}R = \frac{\sum\limits_{c} VP_c}{\sum\limits_{c} VP_c + FN_c} \tag{5.4}$$

onde c representa uma classe (tipo) de entidades nomeadas. Desta maneira, os valores médios são calculados por meio da soma das contribuições de todas as classes, seguida da média deste valor. Isto faz com que o peso de cada entidade nomeada seja o mesmo, independente da classe a qual ela pertence. Consequentemente, entidades nomeadas de classes com maiores representações acabam tendo um maior peso no cálculo deste valor médio.

Outra abordagem para calcular os valores médios de precisão e revocação utiliza as médias de cada classe separadamente, para então calcular o valor médio entre elas. A esta abordagem dá-se o nome de $macro\text{-}P$ e $macro\text{-}R$. Neste trabalho, os valores médios foram calculados utilizando a abordagem *micro*.

Os valores de precisão e revocação, sejam eles valores médios ou apenas para uma classe, podem ser combinados, resultando em uma única métrica. A métrica denominada F-Score [Rijsbergen, 1974] é definida como a média harmônica entre estes valores:

$$F_\beta = (1 + \beta^2)\frac{P \times R}{\beta^2 P + R}. \tag{5.5}$$

Usualmente, define-se $\beta = 1$ e consequentemente a métrica é chamada de F1-Score.

Durante a etapa de avaliação experimental, as métricas anteriormente discutidas foram utilizadas por meio das implementações disponibilizadas pela biblioteca seqeval[1]. Esta biblioteca foi criada com base no script oficial de avaliação do CoNLL-2002. Todas as avaliações foram realizadas da seguinte maneira: para cada ML avaliado, foram realizadas cinco execuções diferentes. A partir dos resultados obtidos pelas execuções, foram calculados os valores médios e seus respectivos desvios padrão.

5.2 Hiperparâmetros

Para o procedimento de *finetuning* do ML baseado na arquitetura BERT, foram mantidas as configurações originais do modelo BERT Multilingual. Utilizou-se o vocabulário original, formado por 119.547 tokens. Este número é consideravelmente alto, em decorrência do pré-treinamento do modelo em dados de idiomas diversos. Para isto, foi utilizado o mesmo processo de tokenização do BERT que é baseado no algoritmo WordPiece [Wu et al., 2016]. Alguns tokens são fragmentados em sub-tokens. Desta forma, é possível representar uma grande quantidade de palavras, inclusive aquelas que estão fora do vocabulário, conforme discutido na Seção 2.4.1. As letras maiúsculas presentes no texto foram preservadas. Para o *finetuning* do ML do BERT no corpus Acórdãos-TCU, foram utilizadas as duas tarefas de treinamento: modelagem de linguagem mascarada e predição de próxima sentença.

O processo de *finetuning* foi realizado durante cinco épocas. Para a taxa de aprendizado, foi definido o valor de 3×10^{-5}. Este valor está próximo do valor utilizado durante o pré-treinamento do ML. Entretanto, existe um limiar de aquecimento para este valor. Durante os primeiros 10 mil passos de treinamento, a taxa de aprendizado é definida em um valor muito próximo a zero e aumenta gradualmente até o valor estabelecido. Esta abordagem é a mesma utilizada no trabalho de Devlin et al. [2018]. O tamanho máximo da sequência foi definido como 256 tokens. Embora esse valor possa atingir 512 tokens,

[1] https://github.com/chakki-works/seqeval

ele causa influência direta no tamanho dos *batches* de treinamento. Por esta razão, optou-se por um tamanho menor, possibilitando utilizar *batches* de treinamento de tamanho 8.

Em relação ao *finetuning* do ML baseado na arquitetura ELMo, utilizou-se como base o modelo ELMo brWaC. O corpus utilizado para o *finetuning*, assim como para o modelo BERT Acórdãos, também foi o Acórdãos-TCU. O *finetuning* foi realizado por uma época apenas, com taxa de aprendizado de 2×10^{-4}. O número de tokens do vocabulário original foi mantido, sendo de 1.516.187. O numero de *batches* de treinamento foi de 128.

Durante a etapa de treinamento supervisionado, foram estabelecidos diferentes quantidades de épocas de treinamento para os modelos baseados na arquitetura BERT. O treinamento supervisionado foi realizado durante 10 épocas no corpora LeNER-Br e HAREM. Além disso, o tamanho máximo das sequências, definido em 512 tokens, e o tamanho dos *batches* de treinamento, definido em 16, foram os mesmos durante o treinamento para os dois corpora. As taxas de aprendizado utilizadas foram de 4×10^{-4} e 4×10^{-5}, respectivamente. Já para o corpus DrugSeizures-Br, foram adotados parâmetros de treinamento diferentes. O treinamento supervisionado foi realizado por 5 épocas. Por se tratar de um corpus consideravelmente maior quando comparado aos demais, o tamanho máximo das sequências foi definido em 256, tanto para o cenário total, quanto para o cenário seletivo. Também foram utilizados *batches* de tamanho 16 e a taxa de aprendizado foi definida em 5×10^{-5}. Também utilizou-se o acumulo de gradientes, que consiste em treinar o modelo em determinada quantidade de *batches* sem atualizar seus parâmetros, enquanto se acumula os gradientes calculados por estes *batches*.

Para os modelos baseados na arquitetura ELMo, o treinamento supervisionado no corpus LeNER-Br foi realizado durante 50 épocas. Contudo, um limiar relacionado ao desempenho do modelo foi estabelecido. Caso não fossem verificadas mudanças durante 25 épocas consecutivas, o treinamento era finalizado. O tamanho definidos para os *batches* de treinamento foi de 32, com taxa de aprendizado de 1×10^{-3}.

5.3 Modelo de Linguagem: Geral X Específico

A primeira parcela dos experimentos realizados é apresentada nesta seção. Nela, foram realizadas avaliações com MLs de diferentes domínios, com o intuito de verificar as diferenças resultantes desta característica, quando avaliados na tarefa de REN. Para isso, são considerados diferentes domínios, tanto para os MLs avaliados quanto para as tarefas de REN.

Na Tabela 5.1 são apresentados os corpora e seus respectivos modelos de

linguagem utilizados durante esta avaliação experimental. Na coluna *ML Geral* são apresentados os dois modelos de linguagem pré-treinados. Para ambos os modelos, foram utilizadas versões pré-treinadas disponibilizadas por trabalhos anteriores, conforme descrito na Seção 3.2. Já na coluna *ML Específico* são descritos os modelos derivados por meio do processo de *finetuning* proposto neste trabalho. Embora as arquiteturas utilizadas sejam diferentes, ambos MLs foram derivados do mesmo corpus específico.

Corpus Geral	**ML Geral**	**Corpus Específico**	**ML Específico**
Wiki Multilingual	BERT Multilingual	Acórdãos-TCU	BERT Acórdãos
brWaC	ELMo brWaC	Acórdãos-TCU	ELMo Acórdãos

Tabela 5.1: Corpora e seus MLs correspondentes. Os MLs Genéricos foram obtidos por meio de pré-treinamento, enquanto os MLs Específicos foram derivados pelo processo de *finetuning*.

Estes modelos foram avaliados por meio das tarefas de REN, conforme apresentado na Tabela 5.2. As tarefas listadas nesta tabela foram avaliadas por diferentes MLs de forma a testar as seguintes hipóteses:

(a) O *finetuning* intradomínio é capaz de melhorar o desempenho dos MLs avaliados em tarefas de domínio específico?

(b) Qual o efeito do *finetuning* intradomínio quando os MLs são avaliados em uma tarefa de domínio geral?

Tarefa de REN		**Modelo de Linguagem**	
Corpus	**Domínio**	**Nome**	**Tipo**
LeNER-Br	Específico	BERT Multilingual	Geral
		BERT Acórdãos	Específico
		ELMo brWaC	Geral
		ELMo Acórdãos	Específico
DrugSeizures-Br	Específico	BERT Multilingual	Geral
		BERT Acórdãos	Específico
HAREM	Geral	BERT Multilingual	Geral
		BERT Acórdãos	Específico

Tabela 5.2: Resumo das avaliações realizadas.

LeNER-Br

A seguir são apresentados os resultados obtidos para as avaliações realizadas no corpus LeNER-Br, de domínio jurídico. Os MLs (gerais e específicos)

listados acima foram utilizados para o treinamento e avaliação no LeNER-Br. Na Tabela 5.3, são exibidos os resultados obtidos pelos diferentes modelos baseados na arquitetura BERT.

BERT - LeNER-Br		
Categoria	Multilingual	Acórdãos
Pessoa	91,38±0,87	93,66±0,21
Tempo	94,43±0,39	92,39±1,82
Local	68,67±1,89	65,78±2,26
Organização	84,76±0,57	86,43±0,49
Legislação	93,40±1,23	95,48±0,47
Jurisprudência	84,04±0,63	83,26±0,73
Total	88,81±0,29	89,39±0,08

Tabela 5.3: Resultados obtidos pelos modelos baseados na arquitetura BERT, avaliados no conjunto de teste do LeNER-Br.

Os resultados obtidos mostram que o modelo BERT Acórdãos, derivado por meio do *finetuning* em dados intradomínio, obteve desempenho superior se comparado ao modelo BERT Multilingual, pré-treinado apenas em dados de domínio geral. Contudo, considerando os desvios padrão, as diferenças observadas não são expressivas.

Embora tenha apresentado desempenho total superior, os resultados alcançados pelo BERT Acórdãos mostram que entidades nomeadas do tipo *Tempo* e *Local* obtiveram os menores resultados se comparados ao modelo BERT Multilingual. Particularmente, os resultados para a entidade do tipo *Local* apresentam valores muito abaixo dos demais. É importante ressaltar que esta classe de entidades representa apenas 3% da entidades nomeadas do conjunto de teste do LeNER-Br. Dessa forma, seu impacto no valor total é baixo.

A maior diferença é observada na entidade nomeada *Tempo*, apresentando dois pontos de diferença em relação ao modelo de base. Já para a entidade nomeada do tipo *Legislação*, observa-se ganho superior a dois pontos de F1-score pelo modelo BERT Acórdãos. Para este tipo de entidade nomeada, não existem muitas variações nas representações das palavras. Como as menções referentes a legislação ocorrem sempre na forma de citações do número da lei ou do seu respectivo artigo, acredita-se que o modelo consegue identificar essas ocorrências sem muitas dificuldades. Estes resultados fortalecem a hipótese de que o *finetuning* do ML em dados intradomínio melhora o desempenho do modelo avaliado na tarefa de REN de domínio específico. A segunda avaliação para o corpus LeNER-Br foi realizada utilizando os modelos baseados na arquitetura ELMo. Na Tabela 5.4 são apresentados os resultados obtidos pelas avaliações. Os resultados remetem ao comportamento observado nas avaliações dos modelos baseados na arquitetura BERT. Esse com-

ELMo - LeNER-Br		
Categoria	brWaC	Acórdãos
Pessoa	95,85±0,76	98,28±0,41
Tempo	91,17±0,89	93,77±1,05
Local	75,04±1,99	75,21±1,13
Organização	86,00±0,82	86,54±0,74
Legislação	89,14±0,82	88,08±1,18
Jurisprudência	80,23±3,88	80,81±1,14
Total	87,66±1,01	88,41±0,41

Tabela 5.4: Resultados obtidos pelos modelos baseados na arquitetura ELMo, avaliados no conjunto de teste do LeNER-Br.

portamento contribui com a hipótese de que o *finetuning* intradomínio de MLs apresenta impacto positivo quando avaliado em tarefas de domínio específico. O modelo ELMo Acórdãos obteve desempenho total superior ao alcançado pelo modelo de domínio geral, ELMo brWaC. Além disso, quando considerado o desempenho por entidades nomeadas, é possível perceber que o modelo ELMo Acórdãos obteve resultado consideravelmente superior para a entidade nomeada *Pessoa*. Por outro lado, para a entidade do tipo *Legislação*, houve uma pequena queda na avaliação em relação ao modelo ELMo brWaC.

5.3.1 DrugSeizures-Br

As avaliações realizadas por meio do corpus DrugSeizures-Brforam conduzidas utilizando apenas os modelos baseados na arquitetura BERT. Na Tabela 5.5 são apresentados os resultados obtidos pelos modelos BERT Multilingual e BERT Acórdãos, avaliados no cenário total do corpus DrugSeizures-Br. É interessante destacar que, diferentemente do que foi observado nos resultados das avaliações no corpus LeNER-Br, o modelo BERT Acórdãos apresentou resultados inferiores quando comparado modelo BERT Multilingual. Avalia-se que o tipo dos documentos que formam o corpus DrugSeizures-Br (petições relacionadas a apreensão de drogas ilícitas (ver Seção 3.4.6)) se diferencia bastante dos documentos que formam o corpus Acórdãos-TCU. Essa característica pode indicar o motivo do desempenho inferior do modelo BERT Acórdãos quando avaliado no DrugSeizures-Br. Ainda que os dois corpora pertençam ao mesmo domínio, essa característica não foi suficiente para garantir que o *finetuning* fosse benéfico. Esse comportamento não foi observado em nenhum cenário das avaliações discutidas anteriormente.

Ainda considerando o corpus DrugSeizures-Br, os modelos propostos também foram avaliados no cenário seletivo, onde diversas entidades nomeadas são condensadas em um menor número de categorias. Para o cenário sele-

BERT - DrugSeizures-Br (Cenário Total)		
Categoria	Multilingual	Acórdãos
Pessoa		
Indiciado	88,21±0,21	86,60±0,14
Testemunha	19,38±2,50	22,03±1,43
Testemunha Ativa	41,74±0,23	38,85±0,42
Testemunha Passiva	0,00±0,00	0,00±0,00
Investigado	0,00±0,00	2,27±0,22
Vítima	30,62±0,27	23,29±1,94
Autor	67,77±0,58	66,20±0,91
Réu	0,00±0,00	0,00±0,00
Advogado	0,00±0,00	0,00±0,00
Tempo		
Tempo	93,42±0,35	92,78±0,14
Data	85,80±0,62	84,18±0,18
Local		
Nome da Rua	61,81±0,40	58,80±0,71
Número da Rua	64,04±0,16	62,67±0,19
Bairro	56,13±0,21	59,44±0,32
Cidade	86,74±0,40	86,39±0,23
Complemento	13,54±11,8	5,97±5,32
Cidade de Origem	68,60±0,26	69,29±0,34
Estado de Origem	56,41±0,44	57,39±0,39
Cidade de Destino	51,27±0,32	52,72±0,16
Estado de Destino	60,39±0,38	62,15±0,14
Organização		
Promotoria	86,47±1,25	88,29±0,20
Autoridade	92,70±0,42	91,50±0,29
Droga		
Nome	89,23±0,79	90,57±0,93
Outro		
Tipo Penal	94,68±2,53	94,07±0,74
Quantidade da Droga	75,98±1,31	74,61±0,35
Total	73,83±1,55	72,78±0,89

Tabela 5.5: Resultados obtidos pelos modelos baseados no BERT, avaliados no conjunto de teste do DrugSeizures-Br (cenário total).

tivo do DrugSeizures-Br, os resultados são exibidos na Tabela 5.6. A partir

BERT - DrugSeizures-Br (Cenário Seletivo)		
Categoria	Multilingual	Acórdãos
Pessoa	81,70±0,31	81,94±0,47
Tempo	87,99±0,11	82,13±0,51
Local	76,80±0,61	77,94±0,65
Organização	92,60±0,58	91,41±0,64
Droga	91,63±0,17	89,71±0,32
Total	80,94±0,07	81,04±0,35

Tabela 5.6: Resultados obtidos pelos modelos baseados na arquitetura BERT, avaliados no conjunto de teste do DrugSeizures-Br (cenário seletivo).

das avaliações, é possível perceber que os resultados do cenário seletivo são superiores aos obtidos no cenário total. Como o número total de categorias de entidade passou de 25 para 5, os modelos podem adaptar-se mais facilmente as categorias, apresentando maior capacidade de generalização. Além disso, no cenário total, algumas categorias de entidades obtiveram resultados nulos ou muito próximos de zero, uma vez que o o número de exemplos era pequeno. Como essas categorias foram agrupadas em supertipos, mesmo que elas sejam preditas erroneamente pelo modelo, os reflexos no resultado final são menores. Entretanto, diferente dos resultados observados no cenário total, o modelo BERT Acórdãos superou o resultado total obtido pelo modelo BERT Multilingual. A entidade nomeada do tipo *Tempo* apresentou a maior diferença de valores entre os dois modelos. Ainda que o modelo BERT Multilingual tenha alcançado uma avaliação quase 6 pontos superior para esta entidade, este modelo foi superado quando considerado o resultado total. Vale ressaltar que a categoria da entidade nomeada *Tempo*, apresenta o menor número de entidades de todo o conjunto. Os resultados obtidos pela avaliação do cenário seletivo do corpus DrugSeizures-Brtambém contribuem com a hipótese levantada inicialmente, considerando que *finetuning* intradomínio de MLs tem impacto positivo quando os modelos são avaliados em corpus de domínio específico.

5.3.2 HAREM

O corpus HAREM também foi utilizado na etapa de avaliação. Dois motivos principais justificam esta escolha. Por ser um corpus muito tradicional, a maioria das avaliações para modelos de REN são realizadas neste corpus. Com essa avaliação, é possível estabelecer parâmetros de comparação em relação aos modelos propostos neste trabalho com outros modelos disponíveis

na literatura. O segundo motivo está relacionado a segunda hipótese levantada anteriormente, que investiga o efeito do *finetuning* intradomínio em MLs avaliados em corpus de domínio geral. Deseja-se avaliar como o *finetuning* intradomínio, mais especificamente em dados do domínio jurídico, é percebido quando aplicado em avaliações generalistas. Na Tabela 5.7 são apresentados os resultados para o HAREM avaliado no cenário total. Os resultados ob-

BERT - HAREM (Cenário Total)		
Categoria	Multilingual	Acórdãos
Pessoa	78,44±0,16	75,49±0,17
Tempo	90,26±0,32	88,67±1,14
Local	81,22±0,41	80,51±0,22
Organização	75,58±0,13	70,24±0,98
Valor	78,96±1,23	76,95±0,32
Obra	48,97±0,76	44,02±3,18
Coisa	38,90±1,78	40,00±2,74
Acontecimento	42,46±2,30	34,67±1,23
Abstração	48,61±3,13	45,05±2,65
Outro	15,98±5,45	12,64±4,23
Total	73,48±0,21	71,03±0,48

Tabela 5.7: Resultados obtidos pelos modelos baseados na arquitetura BERT, avaliados no conjunto de teste do HAREM (cenário total).

tidos mostram que o modelo BERT Acórdãos obteve desempenho total inferior quando comparado ao modelo BERT Multilingual. Isso evidencia que o *finetuning* intradomínio, quando considerados dados do domínio jurídico, prejudica o desempenho dos modelos quando avaliados em tarefas de REN de domínio geral. Embora todos resultados obtidos pelo modelo BERT Acórdãos sejam inferiores, as categorias *Organização* e *Acontecimento* apresentaram as maiores diferenças, sendo de mais de 5 e 7 pontos, respectivamente. Além disso, as categorias *Obra, Coisa, Acontecimento, Abstração* e *Outro* apresentam poucos exemplos, tanto no conjunto de treinamento, quanto no conjunto de teste. Este motivo pode indicar os baixos resultados obtidos para estas classes. As avaliações realizadas para o cenário seletivo do HAREM são apresentadas na Tabela 5.8. O mesmo comportamento apresentado no cenário total foi percebido no cenário seletivo. O modelo BERT Acórdãos apresentou resultados inferiores. Estes resultados reforçam as observações anteriormente destacadas. Embora a diferença percebida entre o resultado total obtido pelos modelos seja menor do que aquela alcançada no cenário total, ainda assim ela contribui para a hipótese de que o *finetuning* intradomínio prejudica o desempenho dos modelos em tarefas de REN de domínio geral.

BERT - HAREM (Cenário Seletivo)		
Categoria	Multilingual	Acórdãos
Pessoa	79,20±1,18	75,91±0,32
Tempo	89,88±0,71	88,22±0,27
Local	79,41±0,25	80,13±0,06
Organização	72,32±0,18	71,29±0,15
Valor	78,34±0,58	75,84±0,54
Total	79,39±0,35	77,72±0,26

Tabela 5.8: Resultados obtidos pelos modelos baseados na arquitetura BERT, avaliados no conjunto de teste do HAREM (cenário seletivo).

5.4 Impacto em Corpora Gerais

Nesta seção é apresentada a segunda parcela das avaliações realizadas por este trabalho. O objetivo destas avaliações é verificar o impacto do *fine-tuning* de MLs, porém considerando apenas corpora de domínio geral. Mais especificamente, deseja-se investigar como as características relacionadas aos corpora utilizados para o *finetuning* tem impacto nestes modelos, quando avaliados na tarefa de REN. O conjunto de avaliações realizadas, bem como os MLs e corpora utilizados são apresentados na Tabela 5.9. Por meio das tarefas descritas na tabela, objetiva-se verificar as seguintes hipóteses:

(a) Qual o impacto do *finetuning* em um corpus genérico mas em português, quando comparado ao Wikipédia Multilingual?

(b) Qual a diferença entre MLs treinados na Wikipédia e no brWaC? Lembrando que o último é bem maior e mais diverso do que o primeiro.

Tarefa de REN		**Modelo de Linguagem**	
Corpus	**Domínio**	**Nome**	**Tipo**
LeNER-Br	Específico	BERT Wikipédia	Geral
DrugSeizures-Br	Específico		
HAREM	Geral		
LeNER-Br	Específico	BERT brWaC	Geral
DrugSeizures-Br	Específico		
HAREM	Geral		
LeNER-Br	Específico	ELMo Wikipédia	Geral

Tabela 5.9: Resumo das avaliações adicionais.

Os MLs utilizados para as avaliações são diferentes daqueles utilizados para as avaliações discutidas nas seções anteriores. O ML BERT Wikipédia foi

obtido por meio do *finetuning* do modelo BERT Multilingual no corpus Wikipédia. Embora o corpus Wikipédia represente uma parcela do corpus utilizado durante o pré-treinamento do modelo BERT Multilingual, o objetivo é verificar o impacto do *finetuning* em um corpus do mesmo idioma das tarefas de REN.

Outro modelo utilizado nas avaliações adicionais foi o BERT brWaC. Este modelo[2] foi pré-treinado no corpus brWaC. Uma vez que os corpora brWaC e Wikipédia, ambos de domínio geral, apresentam diferenças relacionadas ao tamanho e variedade, decidiu-se incluir este modelo nas avaliações realizadas neste trabalho. Desta forma, é possível analisar o impacto destas diferenças nas tarefas de REN em domínio geral e específico. Por último, também foram realizadas avaliações adicionais com um modelo baseado na arquitetura ELMo. Este modelo denominado ELMo Wikipédia foi pré-treinado no corpus Wikipédia.

Inicialmente, os modelos foram avaliados por meio do corpus LeNER-Br. Por simplicidade, as avaliações adicionais são exibidas de forma condensada, concentrando-se apenas nos valores totais de F1-Score obtidos pelos modelos. Além disso, também são exibidos os valores totais obtidos pelos modelos discutidos nas Seções anteriores, de forma a compará-los com os resultados obtidos nesta etapa. Na Tabela 5.10 são ilustrados os resultados dos modelos baseados na arquitetura BERT. A observação dos resultados obtidos pelas

BERT - LeNER-Br	
BERT Multilingual	88.81±0.29
BERT Acórdãos	89.39±0.08
BERT Wikipédia	88.90±1.72
BERT brWaC	90.16±0.37

Tabela 5.10: Resultados totais obtidos pelos modelos baseados na arquitetura BERT, avaliados no conjunto de teste do LeNER-Br.

avaliações adicionais mostra que o resultado obtido pelo modelo BERT Wikipédia está consideravelmente próximo do desempenho alcançado pelo BERT Multilingual. Isto pode estar relacionado à duas causas: Como o modelo BERT Wikipédia foi derivado por meio do *finetuning* do ML BERT Multilingual, utilizando um fragmento do corpus de pré-treinamento, a tendência é que os modelos se comportem de formas semelhante, sem grandes discrepâncias. Dessa forma, o *finetuning* não surtiu grandes efeitos e o modelo BERT Wikipédia se comporta como uma extensão do modelo original. Este comportamento indica que o impacto do processo de *finetuning* neste caso é marginalmente pequeno. Outra possível explicação estaria relacionada ao fato de que, por se tratar de um *finetuning* em domínio distinto ao da tarefa final, a contribuição é ínfima. Por

[2]https://github.com/neuralmind-ai/portuguese-bert

outro lado, o desempenho do modelo BERT brWaC superou todas as outras avaliações. Ainda que o ML tenha sido pré-treinado apenas no corpus brWaC e nenhum processo de *finetuning* tenha sido realizado, ele foi capaz de superar as demais avaliações. Este comportamento pode estar relacionado a natureza do corpus brWaC. Este corpus, proporcionalmente maior que todos os outros corpora utilizados, foi construído utilizando documentos das mais diversas fontes. Dessa forma, a variabilidade linguística do corpus é rica. Essas duas observações sustentam a hipótese de que, o tamanho e a variabilidade do corpus de pré-treinamento, também são capazes de impactar os MLs, quando avaliados por meio da tarefa de REN. Ainda que os documentos do corpus de pré-treinamento sejam de natureza diversa, o impacto percebido foi mais perceptível. Sendo assim, a etapa de pré-treinamento de MLs se estabelece como um fator a ser considerado.

Considerando o modelo baseado na arquitetura ELMo, a única avaliação adicional realizada utilizou o modelo ELMo Wikipédia. Os resultados desta avaliação, bem como os demais resultados discutidos anteriormente são exibidos na Tabela 5.11. Os resultados indicam que o modelo ELMo brWaC,

ELMo - LeNER-Br	
ELMo brWaC	87,66±1,01
ELMo Acórdãos	88,41±0,41
ELMo Wikipédia	86,21±0,68

Tabela 5.11: Resultados totais obtidos pelos modelos baseados na arquitetura ELMo, avaliados no conjunto de teste do LeNER-Br.

pré-treinado no corpus brWaC, supera desempenho do modelo ELMo Wikipédia, pré-treinado no corpus Wikipédia. Ou seja, assim como observado para os modelos baseados na arquitetura BERT, as características do corpus de pré-treinamento causam um impacto nos MLs, quando avaliados na tarefa de REN. É importante ressaltar que o modelo ELMo Acórdãos foi derivado do *finetuning* intradomínio, a partir do modelo ELMo brWaC. Dessa forma, é intuitivo concluir que o desempenho deste modelo seja superior aos demais, considerando que ele foi submetido ao *finetuning* intradomínio.

O corpus DrugSeizures-Br também foi utilizado para as avaliações adicionais. Na Tabela 5.12 são exibidos os resultados obtidos pelas avaliações no corpus DrugSeizures-Br em relação ao cenários total e seletivo. Os resultados mostram que o desempenho do modelo BERT Wikipédia manteve-se próximo aos resultados alcançados pelo modelo BERT Multilingual. Contudo, no cenário total, este desempenho foi ligeiramente inferior. Diferentemente do que foi observado nas avaliações anteriores, o modelo BERT brWaC não obteve resultados superiores a todos os modelos. Quando considerado o cenário total, o

BERT - DrugSeizures-Br		
	Cenário	
ML	Total	Seletivo
BERT Multilingual	73,83±1,55	80,94±0,07
BERT Acórdãos	72,78±0,89	81,04±0,35
BERT Wikipédia	72,98±1,27	81,12±0,18
BERT brWaC	73,19±0,10	80,98±0,07

Tabela 5.12: Resultados obtidos pelos modelos baseados na arquitetura BERT, avaliados no conjunto de teste do DrugSeizures-Br considerando os cenários total e seletivo.

modelo BERT brWaC foi superior ao modelo BERT Acórdãos, o que corrobora com a hipótese relacionada ao impacto causado pelas características do corpus de pré-treinamento. Entretanto, o mesmo comportamento não se repetiu para o cenário seletivo. Neste cenário, o modelo BERT brWaC não foi capaz de superar o modelo com *finetuning* intradomínio, ainda que a diferença seja muito pequena.

Por fim, os modelos também foram avaliados por meio do corpus HAREM e os resultados são exibidos na Tabela 5.13. Nesta avaliação, o modelo BERT

BERT - HAREM		
	Cenário	
ML	Total	Seletivo
BERT Multilingual	73,48±0,21	79,39±0,35
BERT Acórdãos	71,03±0,48	77,72±0,26
BERT Wikipédia	73,85±0,67	79,51±0,54
BERT brWaC	77,70±0,40	83,59±0,14

Tabela 5.13: Resultados obtidos pelos modelos baseados na arquitetura BERT, avaliados no conjunto de teste do HAREM, considerando os cenários total e seletivo.

brWaC atingiu resultados majoritariamente superiores nas avaliações do corpus HAREM, em ambos os cenários. No cenário total a diferença entre o melhor resultado obtido por um modelo proposto neste trabalho, o BERT Wikipédia,e o BERT brWaC foi superior a três pontos. Já no cenário seletivo, a diferença superou quatro pontos. Os resultados observados na Tabela 5.13 contribuem de forma efetiva com a hipótese relacionada ao impacto causado pelo tamanho e variabilidade do corpus utilizado para o pré-treinamento de um ML quando avaliado por meio da tarefa de REN. Todavia, é importante destacar que o HAREM, por ser um corpus de domínio geral, pode tirar o maior proveito de um ML pré-treinado também em corpus de domínio geral.

Além disso, também é menor que os demais corpora avaliados neste trabalho. Dessa forma, o impacto percebido por estas avaliações também pode estar associado à características intrínsecas ao corpus de REN e não somente ao corpus de pré-treinamento do ML.

5.5 Comparação dos Melhores Modelos

Os modelos propostos por este trabalho, cujas avaliações foram discutidas nas seções anteriores, evidenciam o impacto do *finetuning* de MLs por meio da avaliação da tarefa de REN. Na Tabela 5.14 são apresentados os resultados obtidos pelos modelos ELMo Acórdãos e BERT Acórdãos ambos derivados do *finetuning* no corpus Acórdãos-TCU. O resultado total obtido pelo modelo

Categoria	ELMo Acórdãos	BERT Acórdãos
Pessoa	98,28±0,41	93,66±0,21
Tempo	93,77±1,05	92,39±1,82
Local	75,21±1,13	65,78±2,26
Organização	86,54±0,74	86,43±0,49
Legislação	88,08±1,18	95,48±0,47
Jurisprudência	80,81±1,14	83,26±0,73
Total	88,41±0,41	**89,39±0,08**

Tabela 5.14: Comparação dos resultados obtidos pelos modelos derivados do *finetuning* no corpus Acórdãos-TCU, avaliados no conjunto de teste do LeNER-Br.

BERT Acórdãos superou o resultado do modelo baseado no ELMo. Mesmo que resultado total alcançado pelo modelo ELMo Acórdãos tenha sido inferior, percebe-se que categorias como *Pessoa* e *Local*, obtiveram resultados consideravelmente melhores. Para a categoria de entidades *Pessoa*, por exemplo, enquanto o melhor F1-score no BERT foi de 93.66, o ELMo atingiu 98.28. A maior diferença é verificada para a categoria Local, cuja avaliação no BERT atingiu 68.67 contra 75.21 do ELMo, uma diferença de mais de 6 pontos. Por outro lado, ainda que as entidades nomeadas das categorias *Pessoa*, *Tempo* e *Local* tenha alcançado resultados superiores nas avaliações do ELMo Acórdãos, os três tipos representam cerca de 30% de todas as entidades nomeadas do conjunto de teste do LeNER-Br.

Na Tabela 5.15 são exibidos os melhores resultados alcançados pelos modelos avaliados por este trabalho, em comparação com os resultados obtidos por Luz de Araujo et al. [2018], onde o corpus LeNER-Br foi proposto. Para os modelos baseados na arquitetura ELMo, o melhor resultado vem do modelo derivado por meio do processo de *finetuning* intradomínio. Contudo, ainda

Categoria	Luz de Araujo et al. [2018]		ELMo Acórdãos	BERT brWaC
Pessoa	80,83±1,83	(82,14)	98,28±0,41	94,82±0,09
Tempo	90,12±2,28	(89,36)	93,77±1,05	95,26±0,89
Local	64,81±3,05	(66,67)	75,21±1,13	69,41±0,99
Organização	84,37±0,62	(85,48)	86,54±0,74	86,90±0,28
Legislação	93,01±0,61	(94,06)	88,08±1,18	95,63±0,27
Jurisprudência	81,22±1,65	(81,98)	80,81±1,14	82,72±2,61
Total	85,42±0,61	(86,61)	88,41±0,41	**90,16±0,37**

Tabela 5.15: Comparação entre os resultados obtidos pelo trabalho de Luz de Araujo et al. [2018] com os melhores resultados obtidos pelo modelos propostos neste trabalho. São apresentados duas colunas com resultados do modelo proposto por Luz de Araujo et al. [2018]. Na primeira coluna, são apresentados os resultados obtidos por execuções realizadas neste trabalho, usando o código-fonte provido pelos autores. Já os resultados entre parênteses são aqueles produzidos pelos próprios autores.

que o *finetuning* dos MLs em dados intradomínio tenha apresentado um impacto positivo também para os modelos baseados na arquitetura BERT, o melhor resultado obtido nas avaliações do corpus LeNER-Br vem do modelo BERT brWaC. Também são comparados os resultados obtidos neste trabalho, com os resultados descritos no trabalho original do corpus LeNER-Br. Os resultados originais apresentados por Luz de Araujo et al. [2018] foram obtidos por uma avaliação baseada em comparação de palavras, não de entidades. Isso significa que um acerto foi contabilizado a cada palavra de uma entidade predita de forma correta, e não a entidade em sua totalidade. Esse tipo de avaliação não é o padrão utilizado pelos trabalhos disponíveis na literatura. Além disso, essa avaliação potencializa os resultados, com números altos para as métricas utilizadas. Para que fosse possível estabelecer comparações, a mesma avaliação feita em Luz de Araujo et al. [2018] foi executada novamente, porém considerando a métrica de avaliação por entidade. Por meio de cinco execuções distintas, foram obtidos os valores médios seus respectivos desvio-padrão, assim como feito em todas as avaliações deste trabalho. Na Tabela 5.15 são comparados os melhores resultados alcançados na avaliação do LeNER-Br dos modelos resultantes deste trabalho, com a avaliação original do trabalho anteriormente citado. Embora as avaliações do modelo proposto em [Luz de Araujo et al., 2018] tenham sido realizadas novamente para a obtenção dos resultados e seus respectivos desvio-padrão, os resultados originais reportados no trabalho também foram incluídos na Tabela 5.15. Os modelos derivados deste trabalho que apresentaram melhor desempenho nas avaliações no LeNER-Br. Além disso, os resultados obtidos pelo modelo BERT brWaC estabelecem novos valores estado-da-arte para o corpus LeNER-Br.

Também foi avaliado o tempo levado para as etapas de treinamento e pre-

dição, considerando as duas arquiteturas utilizadas. Na Tabela 5.16 são apresentados os tempos de execução para as etapas de treinamento e predição dos modelos. Para estabelecer a avaliação em relação ao tempo de treinamento e

	BERT	ELMo
Treinamento	4393,20±4,49	1842,75±65,10
Validação	22,04±0,12	156,23±00,25

Tabela 5.16: Tempo de execução em segundos do BERT e ELMo em segundos. Treino considerado durante 10 épocas em ambos os modelos.

predição, utilizou-se tamanho de *batch* que fosse máximo para ambos modelos. É evidente a diferença entre o tempo de treinamento do BERT em relação ao ELMo. Contudo, esta diferença não é percebida para a predição. O tempo de predição do BERT é cerca de sete vezes menor quando comparado com o ELMo. Estas avaliações de tempo foram realizadas em uma máquina equipada com uma GPU do tipo GeForce GTX 1080 Ti que possui 11 GB de memória e 3.584 núcleos.

CAPÍTULO

6

Conclusões

Neste trabalho foram avaliados os impactos causados pelo *finetuning* de modelos profundos de linguagem por meio da tarefa de Reconhecimento de Entidades Nomeadas aplicada em corpora de domínio geral e específico. Utilizando o processo de *finetuning*, foram derivados novos modelos, que posteriormente foram utilizados para o treinamento supervisionado utilizando corpus anotados para a tarefa de REN. Finalmente, após o treinamento supervisionado, os modelos foram submetidos a avaliações realizadas pelas métricas definidas para a tarefa específica. O principal objetivo das avaliações realizadas consistia em verificar o impacto causado pelo *finetuning* dos MLs, considerando dados de domínio geral e específico, tanto na etapa de *finetuning* quanto na avaliação da tarefa de REN. Além disso, o trabalho também propõe um novo corpus dedicado a tarefa de REN, formado por petições submetidas ao Ministério Público do Estado de Mato Grosso do Sul, com entidades nomeadas manualmente anotadas. A este corpus deu-se o nome de DrugSeizures-Br.

Neste capítulo são apresentadas as conclusões deste trabalho. Na Seção 6.1 é realizado um paralelo entre os objetivos desta tese e os resultados obtidos. Na Seção 6.3 são discutidas algumas limitações das soluções propostas e na Seção 6.4 são apresentadas algumas direções de trabalhos futuros.

6.1 Resumo dos Objetivos e Principais Resultados

A principal motivação do trabalho, a avaliação do impacto do *finetuning* de modelos de linguagem profundos em dados intradomínio aplicados a tarefa de REN foi alcançada por meio dos métodos descritos na Seção 3. Pela análise dos resultados obtidos nos experimentos realizados, foram estabelecidas as

seguintes conclusões:

- O *finetuning* de modelos profundos de linguagem em dados de domínio geral resulta em um impacto consideravelmente pequeno, quando avaliado por meio da tarefa de REN, tanto em corpora de domínio geral e jurídico. Este impacto foi avaliado em relação as diferenças existentes nos corpora utilizados no *finetuning*.
- O *finetuning* de modelos profundos de linguagem em dados do domínio jurídico resulta em um impacto positivo, quando avaliado por meio da tarefa de REN em corpora de domínio jurídico. Contudo, além do domínio, outras características relacionadas a natureza dos textos que formam os corpora, como a variabilidade linguística e o estilo de escrita, também podem causar influências. Por outro lado, o impacto é negativo quando os mesmo modelos são aplicados à tarefa de REN em corpora de domínio geral.
- A etapa de pré-treinamento de modelos profundos de linguagem também pode influenciar o desempenho destes modelos avaliados por meio da tarefa de REN. Características relacionadas ao tamanho e variabilidade do corpus utilizado nesta etapa podem ser determinantes.
- Modelos baseados na arquitetura BERT mostraram-se superiores nas avaliações realizadas no corpus LeNER-Br. Mesmo os modelos de base apresentaram resultados superiores.

6.2 Contribuições

As principais contribuições resultantes deste trabalho são:

- modelos de linguagem profundos com *finetuning* realizado em dados do domínio jurídico;
- avaliação da técnica de finetuning intra-domínio para REN em domínio jurídico; e
- resultados estado-da-arte para o corpus LeNER-BR.

6.3 Limitações

Algumas das limitações encontradas durante a execução deste trabalho estão relacionadas a escassez de recursos de PLN dedicados ao idioma Português. O baixo número de corpora anotados disponíveis na literatura condiciona as avaliações a um mesmo corpus. Além disso, também são poucos os

modelos pré-treinados dedicados ao Português. Contudo, durante a execução deste trabalho, a disponibilização de um modelo pré-treinado baseado na arquitetura do BERT foi um importante acontecimento. Outra limitação encontrada está relacionada à capacidade de processamento computacional. Modelos baseados em aprendizado profundo exigem alto poder de processamento. Embora a estrutura utilizada durante o desenvolvimento tenha atendido aos objetivos aqui propostos, a disponibilidade de infraestrutura computacional mais potente e robusta é um fator que pode ser determinante na deste tipo de trabalho.

6.4 Trabalhos Futuros

O uso de modelos de linguagem profundos voltados para o Português ainda apresenta muitas áreas inexploradas. A partir dos resultados obtidos neste trabalho, é possível propor a continuação do trabalho como segue:

- Pré-treinamento de modelos de linguagem profundos utilizando apenas dados do domínio jurídico;
- *Finetuning* de modelos de linguagem profundos utilizando documentos do domínio jurídico, porém obtidos de fontes distintas;
- Avaliação de modelos baseados em outras arquiteturas de aprendizado profunda, como o GPT [Radford et al., 2018] e o T5 [Raffel et al., 2019].

Bibliografia

N. Cardoso. Harem e miniharem: Uma análise comparativa, 2006. Citado nas páginas 2 e 33.

P. Castro, N. Felix, and A. Soares. Portuguese named entity recognition using lstm-crf. In *Proceedings of the Computional Processing of the Portuguese Language*, 07 2018. Citado nas páginas 25, 33, e 41.

T. Chakrabarty, C. Hidey, and K. McKeown. IMHO fine-tuning improves claim detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 558–563, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1054. URL https://www.aclweb.org/anthology/N19-1054. Citado na página 42.

I. Chalkidis and I. Androutsopoulos. A deep learning approach to contract element extraction. In *JURIX*, 2017. Citado na página 43.

J. P. C. Chiu and E. Nichols. Named entity recognition with bidirectional lstm-cnns. *CoRR*, abs/1511.08308, 2015. URL http://arxiv.org/abs/1511.08308. Citado nas páginas 40 e 41.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. Natural language processing (almost) from scratch. *CoRR*, abs/1103.0398, 2011. URL http://arxiv.org/abs/1103.0398. Citado na página 40.

A. Cucchiarelli and P. Velardi. Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Computational Linguistics*, 27 (1):123–131, 2001. URL https://www.aclweb.org/anthology/J01-1005. Citado na página 39.

A. M. Dai and Q. V. Le. Semi-supervised sequence learning. *CoRR*, abs/1511.01432, 2015. URL http://arxiv.org/abs/1511.01432. Citado na página 42.

P. V. Q. de Castro. Aprendizagem profunda para reconhecimento de entidades nomeadas em domínio jurídico. Master's thesis, Programa de Pós-Graduação em Ciência da Computação - Instituto de Informática (INF) - Universidade Federal de Goiás (UFG), 2019. Citado nas páginas 25 e 44.

P. V. Q. de Castro, N. F. F. da Silva, and A. da Silva Soares. Contextual representations and semi-supervised named entity recognition for portuguese language. In *IberLEF@SEPLN*, 2019. Citado na página 43.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805. Citado nas páginas 1, 9, 25, 30, e 47.

C. N. dos Santos and V. Guimarães. Boosting named entity recognition with neural character embeddings. *CoRR*, abs/1505.05008, 2015. URL http://arxiv.org/abs/1505.05008. Citado na página 33.

J. F. dos Santos Neto. Reconhecimento de entidades nomeadas para o português usando redes neurais. Master's thesis, Programa de Pós-Graduação em Ciência da Computação - Pontifícia Universidade Católica do Rio Grande do Sul (PUC-RS), 2019. URL http://tede2.pucrs.br/tede2/handle/tede/9050. Citado na página 33.

C. Dozier, R. Kondadadi, M. Light, A. Vachher, S. Veeramachaneni, and R. Wudali. Named entity recognition and resolution in legal text. In *Semantic Processing of Legal Texts*, pages 27–43. Springer, 2010. Citado na página 43.

I. Fernandes, H. Lopes Cardoso, and E. Oliveira. Applying deep neural networks to named entity recognition in portuguese texts. In *Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS*, pages 284–289, 10 2018. doi: 10.1109/SNAMS.2018.8554782. Citado na página 41.

F. Galgani, P. Compton, and A. Hoffmann. Combining different summarization techniques for legal text. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, HYBRID '12, page 115–123, USA, 2012. Association for Computational Linguistics. Citado na página 43.

R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, page 466–471, USA, 1996. Association for Computational Linguistics. doi: 10.3115/992628.992709. URL https://doi.org/10.3115/992628.992709. Citado na página 5.

J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 267–274, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: 10.1145/1571941.1571989. URL http://doi.acm.org/10.1145/1571941.1571989. Citado na página 6.

S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith. Don't stop pretraining: Adapt language models to domains and tasks, 2020. Citado na página 43.

X. Han and J. Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 215–224, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585123. doi: 10.1145/1645953.1645983. URL https://doi.org/10.1145/1645953.1645983. Citado na página 39.

N. Hartmann, E. R. Fonseca, C. Shulby, M. V. Treviso, J. S. Rodrigues, and S. M. Aluísio. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *CoRR*, abs/1708.06025, 2017. URL http://arxiv.org/abs/1708.06025. Citado na página 31.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735. Citado na página 40.

J. Howard and S. Ruder. Universal language model fine-tuning for text classification. *CoRR*, abs/1801.06146, 2018. URL http://arxiv.org/abs/1801.06146. Citado nas páginas 1, 42, e 43.

Z. Huang, W. Xu, and K. Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015. URL http://arxiv.org/abs/1508.01991. Citado nas páginas 40 e 41.

C. M. Júnior, H. Maced, T. Bispo, F. Santos, N. Silva, and L. Barbosa. Paramopama: a brazilian-portuguese corpus for named entity recognition. *XII Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, 2015. Citado na página 34.

M.-Y. Kim and R. Goebel. Two-step cascaded textual entailment for legal bar exam question answering. In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ICAIL '17, page 283–290, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348911. doi: 10.1145/3086512.3086550. URL https://doi.org/10.1145/3086512.3086550. Citado na página 43.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781. Citado na página 31.

G. Lample and A. Conneau. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291, 2019. URL http://arxiv.org/abs/1901.07291. Citado na página 1.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural architectures for named entity recognition. *CoRR*, abs/1603.01360, 2016. URL http://arxiv.org/abs/1603.01360. Citado nas páginas 31 e 40.

Y. LI, K. BONTCHEVA, and H. CUNNINGHAM. Adapting svm for data sparseness and imbalance: a case study in information extraction. *Natural Language Engineering*, 15(2):241–271, 2009. doi: 10.1017/S1351324908004968. Citado na página 45.

P. H. Luz de Araujo, T. E. de Campos, R. R. R. de Oliveira, M. Stauffer, S. Couto, and P. Bermejo. LeNER-Br: a dataset for named entity recognition in Brazilian legal text. In *International Conference on the Computational Processing of Portuguese (PROPOR)*, Lecture Notes on Computer Science (LNCS), pages 313–323, Canela, RS, Brazil, September 24-26 2018. Springer. doi: 10.1007/978-3-319-99722-3_32. URL https://cic.unb.br/~teodecampos/LeNER-Br/. Citado nas páginas xiv, 2, 8, 34, 59, e 60.

X. Ma and E. H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *CoRR*, abs/1603.01354, 2016. URL http://arxiv.org/abs/1603.01354. Citado na página 41.

C. Mota and D. Santos, editors. *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*. Linguateca, 2008. URL http://www.linguateca.pt/LivroSegundoHAREM/. ISBN: 978-989-20-1656-6. Citado na página 33.

L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin. How transferable are neural networks in NLP applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1046. URL https://www.aclweb.org/anthology/D16-1046. Citado na página 1.

O. Muñoz, A. Pomares Quimbaya, A. Sierra, R. Gonzalez, and A. García. Named entity recognition over electronic health records through a combined dictionary-based approach. volume 100, pages 55–61, 10 2016. doi: 10.1016/j.procs.2016.09.123. Citado na página 39.

M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202. Citado nas páginas 1 e 19.

M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power. Semi-supervised sequence tagging with bidirectional language models. *CoRR*, abs/1705.00108, 2017. URL http://arxiv.org/abs/1705.00108. Citado na página 9.

J. Pirovani and E. Oliveira. Portuguese named entity recognition using conditional random fields and local grammars. In *Proceedings of LREC'18*, May 2018. Citado na página 41.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018. Citado nas páginas 1 e 65.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019. Citado na página 65.

L. Ramshaw and M. Marcus. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, 1995. URL https://www.aclweb.org/anthology/W95-0107. Citado nas páginas 7 e 35.

V. Rijsbergen. Foundation of evaluation. *Journal of Documentation*, 30, 1974. URL https://doi.org/10.1108/eb026584. Citado na página 47.

K. Rother and A. Rettberg. Ulmfit at germeval-2018: A deep neural language model for the classification of hate speech in german tweets. 09 2018. Citado na página 42.

D. Santos and N. Cardoso. *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*. Linguateca, 2007. Citado nas páginas 2 e 33.

F. Souza, R. Nogueira, and R. Lotufo. Portuguese named entity recognition using bert-crf, 2019. Citado nas páginas 30, 33, e 44.

I. Sutskever, O. Vinyals, and Q. Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 4, 09 2014. Citado na página 42.

W. L. Taylor. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433, 1953. doi: 10.1177/107769905303000401. URL https://doi.org/10.1177/107769905303000401. Citado na página 16.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL https://www.aclweb.org/anthology/W03-0419. Citado na página 34.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762. Citado nas páginas 9, 12, e 15.

J. A. Wagner Filho, R. Wilkens, M. Idiart, and A. Villavicencio. The brWaC corpus: A new open resource for Brazilian Portuguese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L18-1686. Citado na página 26.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. Citado nas páginas 11 e 47.

V. Yadav and S. Bethard. A survey on recent advances in named entity recognition from deep learning models, 2019. Citado na página 6.

G. Zhou and J. Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 473–480, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073163. URL https://doi.org/10.3115/1073083.1073163. Citado na página 40.