Avaliação de modelos de DeepLearning para Reconhecimento de Entidades Nomeadas no Domínio Legal em Língua Portuguesa

Hidelberg O. Albuquerque[1]
[1]Centro de Informática, Universidade Federal de Pernambuco, Recife, Brasil
Email: hoa@cin.ufpe.br

Resumo*— *A cada ano, uma grande quantidade de documentos jurídicos e legislativos é gerada, dificultando a atuação de profissionais da área. A maioria dos dados produzidos não possui padronização e o uso de ferramentas de aprendizagem profundo pode minimizar atrasos e acelerar processos. Sistemas que utilizam Deep Learning com Reconhecimento de Entidades Nomeadas (NER) possuem potencial para a extração das informações em documentos legais com eficácia. Este trabalho foram avaliados dois modelos de aprendizado associados às tarefas de NER, BI-LSTM+CRF e BERT+Fine-Tunning, buscando reproduzir o estado-da-arte e melhorar resultados experimentais através de otimizações nos hiperparâmetros dos modelos. Os resultados obtidos demonstraram o que o modelo BERT foi superior ao outro modelo, além de conquistar resultado médio melhor que o do artigo-base.

Keywords—Named Entity Recognition. Aprendizado Profundo. Benchmark.

I. INTRODUÇÃO

Processamento de Linguagem Natural (*Natural Language Processing*– NLP) é uma subárea interdisciplinar da inteligência artificial que abrange vários campos de estudo, como inteligência computacional e linguística. A área de NLP é um desafio para pesquisadores e profissionais pois corresponde em como a linguagem natural, geralmente escrita de forma não-estruturada e repleta de riquezas e complexidades, pode ser transformada e utilizada por sistemas computacionais [1]. O processo de estruturação das informações é fundamental para o NLP, pois à medida que as informações são estruturadas elas passam a ser mais facilmente indexadas e interpretadas. Entre as técnicas existentes, o Reconhecimento de Entidades Nomeadas (*Named Entities Recognition* - NER) visa identificar entidades no texto e classificá-las em determinados conjuntos de categorias semânticas genéricas como nomes de pessoas, localização, organizações [2], ou específicas de uma determinada língua e/ou de um determinado domínio. Dada à variedade semântica das informações em línguas/domínios distintos, é importante determinar quais métodos pode ser melhor empregados. Estas particularidades indicam possíveis razões pelas quais muitas das pesquisas focam numa abordagem monolinguística, principalmente em textos escritos na língua inglesa [3, 4]. Pesquisas utilizando língua portuguesa precisam de um maior empenho, principalmente no desenvolvimento de abordagens e ferramentas de alta qualidade, como ocorre em outras línguas [5, 6].

Por ser uma área multidisciplinar, as tarefas associadas à NLP são utilizadas em outros domínios, tais como Geologia [5], Jornalismo [7] e Biomedicina [8], por exemplo. Em domínios específicos, estas técnicas possibilitam extrair informações particularmente relevantes. Na literatura encontram-se estudos relacionados ao uso de aprendizagem de máquina dentro do domínio jurídico [9-12]. O domínio jurídico ou domínio legal inclui uma grande variedade de textos, como leis, projetos de lei, processos legais, acórdãos, comunicações oficiais, entre outros. Uma imensurável quantidade de documentos jurídicos ou legislativos é gerada anualmente, o que tem dificultado a atuação de advogados, juízes, legisladores e pesquisadores. A maioria dos dados produzidos é definida em textos livres sem padronização. O entendimento mais eficaz destes dados é um desafio que pode ser auxiliado com a ajuda da tecnologia através da atribuição semântica e significância mais precisas das informações encontradas nos documentos legais, minimizando atrasos na busca ou na análise de correspondências jurídicas, entre outras vantagens [13-15]. Os dados classificados e o conjunto de *features* extraídos podem ser utilizados para treinamento de modelos de aprendizado profundo, possibilitando, por exemplo, a diminuição do conjunto de referências encontradas, diminuindo conflitos e minimizando empecilhos processuais. Apesar de existirem pesquisas voltadas para o domínio do Direito, existem poucos trabalhos voltados para a língua portuguesa e para a justiça brasileira, dentre os quais destacamos os trabalhos de Luz de Araújo et al [9] e Quinta de Castro [16].

Redes Neurais Profundas (*Deep Neural Networks*) são um conjunto de métodos de aprendizado de máquina aplicados em diferentes domínios, como visão computacional e tarefas de NLP, entre outros [17]. Modelos de aprendizado baseados em Aprendizado Profundo (*Deep Learning - DL*) nas tarefas de NLP têm obtido resultados relevantes em tarefas como sumarização de documentos, reconhecimento de fala, análise de sentimento, sistemas de pergunta-resposta e NER. Nas tarefas de NER, técnicas de DL são capazes de detectar, identificar e classificar entidades com desempenho superior às técnicas tradicionais de aprendizagem de máquina [18]. Neste cenário, a otimização de hiperparâmetros objetiva melhorar o resultado final de modelos de DL, combinando um conjunto de possibilidades em um espaço de configurações [19]. A escolha adequada dos hiperparâmetros do modelo pode influenciar sensivelmente o desempenho e o resultado final d aprendizado [20]. Esta escolha possui alguns desafios, como o custo computacional na avaliação de muitos conjuntos de combinações, alta dimensionalidade e complexidade do espaço de configurações, tempo de treinamento e outros [21].

Neste trabalho, descrevemos o processo de benchmark de duas arquiteturas de DL, Bi-LSTM+CRF e BERT, aplicados às tarefas de NER no contexto jurídico/legislativo. Para tanto, foram utilizados dois corpora públicos deste domínio [9, 22], buscando inicialmente alcançar os resultados obtidos no estado da arte e apontar o melhor desempenho em termos

de Acurácia, Precisão, Recall e F1-score, métricas amplamente utilizadas na academia, a partir da combinação de hiperparâmetros dos modelos. O restante do trabalho está organizado da seguinte maneira: a Seção II apresenta brevemente os trabalhos relacionados ao estado-da-arte. A Seção III apresenta a proposta desta pesquisa. A Seção IV descrimina as etapas dos experimentos. A Seção V apresenta os resultados e as discussões obtidas. Por fim, a Seção VI apresenta as conclusões e os trabalhos futuros.

II. Trabalhos Relacionados

O uso de técnicas de NER para o domínio jurídico tem sido explorado em diferentes línguas e domínios [6, 13, 23-26]. No entanto, os corpora públicos são escassos para o domínio legal em português. Esta seção destaca contribuições para este domínio e linguagem.

Luz de Araújo et al. [9] apresentou o "LeNER-Br", um corpus obtido de 70 documentos jurídicos de tribunais e leis brasileiras, utilizando entidades já definidas no corpus HAREM [27], acrescentando novas entidades para extrair conhecimento jurídico: "Legislação" para leis e "Jurisprudência" para decisões legais resultantes de processos judiciais. Utilizando as arquiteturas Bi-LSTM+CRF e o vetor de palavras *Glove* pré-treinado em português brasileiro e europeu, os autores alcançaram F1-score de 88,82% para casos jurídicos e 97,04% para legislação. Os códigos e o corpus anotados utilizados estão disponíveis publicamente na página dos autores [1]. Sua pesquisa foi considerada pioneira para tarefas de NER no contexto jurídico brasileiro, sendo referenciada como artigo-base pela comunidade científica.

Alles [28] desenvolveu o "DOU-Corpus" para NER a partir de 470 documentos feitos no Diário Oficial da União (DOU), o boletim oficial de publicações administrativas do Governo Federal Brasileiro. Novas entidades foram usadas para o domínio legal, como "Cargo" (ocupação profissional), "Lei", "Numero", "Processo" e "Valor-monetário". Utilizando a ferramenta Apache OpenNLP[2], os resultados foram avaliados quantitativa e qualitativamente. A precisão, cobertura e F-score do modelo foram, respectivamente, 95,3%, 60,7% e 44,5%. Não foi encontrado o corpus e o código utilizados.

Castro [16] desenvolveu um modelo de NER para a língua portuguesa no domínio da Justiça do Trabalho no Brasil. O autor inicialmente propôs melhorar a precisão dos modelos NER para a língua portuguesa usando arquiteturas e representações de palavras baseadas em *Deep Neural Networks*. Foram utilizados 1.305 documentos, nos quais as anotações das entidades foram realizadas de forma semisupervisionada, utilizando entidades clássicas do estado da arte e criando as entidades para o contexto jurídico: "Função", "Fundamento" (disposições legais), "Tribunal " e "Vara" (características das organizações jurídicas brasileiras), além de "Valor Acordo", "Valor Causa", "Valor Condenação" e "Valor Custas" (para diferentes tipos de valores em ações trabalhistas). A rede Bi-LSTM+CRF obteve resultados acima de 80% de precisão. Não foi encontrado o corpus e o código utilizados.

Luz de Araújo, et al. [12] apresenta "VICTOR", um dataset construído a partir de documentos do Supremo Tribunal Federal, composto por mais de 692 mil documentos anotados manualmente por especialistas. Além do conjunto de dados em si, os autores apresentam como contribuições a classificação dos dados em dois tipos de tarefas: tipo de documento, fazendo distinção entre os tipos de documentos jurídicos, e classificação do tema do processo. Utilizando os modelos CNN e Bi-LSTM+CRF, os experimentos mostram que a natureza sequencial dos processos pode ser aproveitada para melhorar a classificação dos tipos de documentos. Os datasets anotados não foram encontrados.

Albuquerque et al [22] apresenta o "UlyssesNER-Br", um corpus para tarefas de NER desenvolvido a partir de documentos legislativos. A pesquisa foi realizada no contexto do Projeto Ulysses, um conjunto institucional de iniciativas de inteligência artificial da Câmara dos Deputados Brasileira. O trabalho apresenta a análise de dois corpora, "PL-Corpus", desenvolvido a partir de 150 documentos públicos com projetos de lei disponíveis no portal da Câmara, e o "ST-corpus", com 795 documentos com solicitações de trabalho internas fornecidos pela Câmara. Os documentos foram anotados manualmente por três equipes de anotadores, em três fases distintas. Por possui dados sigilosos, o ST-corpus não está disponível publicamente; todavia os dados anotados do PL-corpus estão disponíveis para utilização [3]. Além das entidades mais comuns, o UlyssesNER-Br introduz as entidades para o domínio legislativo "Fundamento" (para leis, decretos, emendas, etc.), e "Produto de Lei" (para produtos gerados por leis). Utilizando os modelos de aprendizagem de máquina Hidden Markov Model (HMM) e Conditional Random Fields (CRF), foi alcançado o resultado de 97.27% de Acurácia e 76.28% de F1-score utilizando a técnica de validação cruzada.

III. Proposta

Esta pesquisa se propôs avaliar o desempenho de duas arquiteturas de Deep Learning: Bi-LSTM+CRF disponibilizada por Luz de Araújo et al. [9], e um modelo BERT para arquitetura Transformers pré-treinado para o domínio legal [4]. O primeiro modelo de aprendizado foi escolhido devido a sua expressividade no estado-da-arte. O modelo BERT têm apresentado melhorias consideráveis nos resultados para tarefas de NER [30, 31].

Para identificação de entidades, foram utilizados dois corpora: o Lener-Br corpus (artigo-base), devido seu pioneirismo e sua aceitação como referencial no estado-da-arte; e o PL-Corpus, devido a sua similaridade semântica com o corpus do artigo-base. Além disto, outros critérios utilizados foram disponibilidade de acesso aos corpora, e a anotação rotulada utilizando o padrão Conll2002 [32].

Utilizando cunho exploratório, foi utilizada uma metodologia experimental, inicialmente aplicando os modelos aos corpora selecionados, buscando reproduzir os resultados do estado da arte. Após esta fase inicial, foram aplicadas otimizações nos hiperparâmetros dos modelos, buscando melhorar o desempenho nas tarefas de NER. Assim, buscou-se responder a seguinte pergunta: "das arquiteturas escolhidas, qual possui o melhor resultado em

[1] *https://github.com/peluz/lener-br*
[2] *https://opennlp.apache.org*
[3] *https://github.com/bergoliveira/disciplinaDL/tree/main/pl_corpus*
[4] *https://huggingface.co/pierreguillou/ner-bert-base-cased-pt-lenerbr*

TABLE I.
HIPERPARÂMETROS DE TREINAMENTO ORIGINAIS (ARTIGOS-BASE)

Hiperparâmetros	Modelos		Espaço de configurações[1]
	Bi-LSTM+CRF	BERT+Fine tunning	
Word Embedings	*Glove*	--	[Glove, Word2Vec, FastText, Wang2Vec][2]
Número de Batchs	10	10	[10,20,40]
Épocas (ambos)	55	55	[35,55,75]
Método de otimização	SGD	AdamW	[SGD, Adam, Adagrad, RMSProp][3]
Taxa de aprendizado	0,015	2e-5	[1e-4, 2e-5, 3e-4, 5e-5][4]

[1]Espaço de configurações utilizadas para os experimentos. [2]O modelo BERT escolhido não utiliza *word embbedings*. [3]Não foi possível observar no modelo BERT a possibilidade de alteração de métodos de otimização. O impacto de Word embbedings e Métodos de Otimização foram testados somente no modelo Bi-LSTM+CRF. [4] Nos experimentos com o Bi-LSTM+CRF, este parâmetro não foi alterado, por ter sido avaliados métodos de otimização que utilizavam taxas de aprendizado variado. Sua influência foi verificada somente no modelo BERT.

TABLE II.
EXPERIMENTOS INICIAIS UTILIZANDO O LENER-BR CORPUS[1].

Modelo[2]	Entidades	Resultados							
		Artigo-base[3]				Resultado-base[4]			
		AC	PR	RC	F1	AC	PR	RC	F1
Bi-LSTM+CRF com Glove	Pessoa	---	94,4	92,5	93,4	---	94,1±5,3	93,6±8,1	**93,8±7,0**
	Local	---	61,2	59,8	**60,5**	---	45,4±12,2	55,8±14,3	49,3±11,6
	Organização	---	91,2	85,6	**88,3**	---	83,5±6,6	74,0±14,1	78,0±12
	Tempo	---	91,1	91,1	**91,1**	---	86,5±1,8	88,5±3	87,4±1,8
	Legislação	---	97,0	97,0	**97,0**	---	86,4±4,6	85,8±8,8	86,0±7,2
	Jurisprudência	---	87,3	90,3	**88,8**	---	68,4±10,8	64,9±12,8	66,2±12,0
	Overall[6]	(ni)	93,2	91,9	**92,5**	96,3±1,5	82,3±3,3	78,5±10,1	80,1±7,8
BERT+Fine Tunning	Pessoa	--	98,7	97,7	98,2	---	99,4±4,8	98,3±7,6	**98,8±6,5**
	Local	--	73,6	93,8	**82,5**	---	76,7±11,7	86,6±13,9	81,3±11,1
	Organização	--	91,8	86,9	**89,3**	---	89,1±6,1	89,0±13,6	89,1±11,5
	Tempo	--	94,7	98,5	96,5	---	94,4±1,2	98,5±2,5	**96,9±1,2**
	Legislação	--	89,4	87,3	**88,4**	---	88,0±3,4	83,7±7,5	85,8±6
	Jurisprudência	--	64,2	77,3	70,1	---	76,3±10,1	73,5±12,1	**74,8±11,3**
	Overall	97,26	88,10	90,45	**89,26**	96,9±0,3	86,3±2,8	88,7±1,5	87,5±1,6

[1]Métricas: AC: Acurácia, PR: Precisão, RC: Recalll e F1: F1-score. Os resultados estão em percentuais (%) [2]Para os experimentos iniciais, não foram feitas alterações nos códigos ou nos hiperparâmetros originais dos modelos.. [3]Nos resultados do artigo-base, não foram encontradas informações sobre a acurácia geral do modelo. Os modelos retornavam somente acurácia geral. [4]Os resultado-base apresentados representam a média±desvio-padrão obtidos após cinco execuções. Os resultados-base serão considerados como referência para os demais experimentos.

TABLE III.
EXPERIMENTOS INICIAIS UTILIZANDO O PL-CORPUS

Entidades	Resultados											
	Artigo Base[1] Conditional Random Fields (CRF)[2]				Resultado-base[3] Bi-LSTM+CRF com Glove				Resultado-base[3] BERT+FineTunning			
	AC	PR	RC	F1	AC	PR	RC	F1	AC	PR	RC	F1
Pessoa	--	82,6	51,1	63,1	--	78,6±2.8	67.9±7.8	72.6±5.7	--	91,5±1	91,1±1,8	**91,3±0,8**
Local	--	72,5	71,0	71,7	--	74,2±3.7	74.3±16.9	72.9±14.4	--	80,8±1,4	83,8±2,3	**82,3±0,5**
Organização	--	66,2	51,2	57,7	--	70,1±10.2	54.1±16.8	60.1±16.4	--	89,9±3	78,8±3,4	**81,7±3,1**
Data	--	88,6	68,2	77,1	--	87.1±3.2	90.7±3.6	88.8±3.31	--	84,9±5,9	96,8±2,1	**90,4±3,0**
Evento	-	0,0	0,0	0,0	--	80.9±36.9	31.1±14	44.8±20.1	--	100±18,9	62,5±17,5	**76,9±18**
Fundamento	--	67,6	75,3	71,2	--	76.4±7.7	75.6±8.5	75.9±8.1	--	77,2±2,3	81,9±5,2	**79,5±3,2**
Produto de Lei	--	74,1	31,8	44,5	--	41.0±16.4	26.8±9.6	32.2±12.3	--	49,4±2,5	54,4±3,7	**51,8±4,0**
Overall	97,2±0,7	83,4±0,9	70,4±1,5	76,2±1,1	94,2±8,8	73,1±10,5	68,4±11,0	70,6±10.7	97,1±0,3	81,1±2,8	83,3±1,5	**82,2±1,6**

[1]Métricas: AC: Acurácia, PR: Precisão, RC: Recalll e F1: F1-score. Os resultados estão em percentuais (%).[2] Os resultados do artigo-base não apresentam os desvios padrão por entidade, nem acurácia por entidade. Por não utilizar um modelo de deep learning, não foram utilizados os códigos do artigo-base, somente os resultados e o corpus. [3] Experimentos iniciais aplicando os modelos anteriormente utilizados [19,33], sem alterações nos códigos ou hiperparâmetros originais. Os resultado-base apresentados representam a média±desvio-padrão obtidos após cinco execuções. Os resultados-base serão considerados como referência para os demais experimentos neste trabalho.

termos de Acurácia, Precisão, Recall e F1-score nas tarefas de NER em textos legais?".

IV. EXPERIMENTOS

Inicialmente, para se ter um objeto de comparação, foi aplicado os modelos Bi-LSTM+CRF e BERT dos artigos-base nos corpora disponibilizados. Os datasets utilizados na etapa de Fine-Tunning no modelo BERT estão disponíveis na plataforma Hugging Face[5]. Não houve alterações no código e nos hiperparâmetros originais, demonstradas na Tabela I. Não foi encontrada no Lener-Br a quantidade de execuções efetuadas. Assim, para se ter um resultado médio, neste trabalho foram feitas cinco execuções e calculadas a média e desvio-padrão geral e por entidade, utilizando a métrica F1-score como determinante para o resultado final.

[5] *https://huggingface.co/datasets/lener_br*
https://huggingface.co/datasets/bergoliveira/pl_corpus

As Tabelas II e III mostram os resultados iniciais, com destaque em negrito para melhores *scores*.

Os resultados dos experimentos iniciais obtidos utilizando os modelos selecionados nos corpora Lener-Br e PL-corpus não conseguiram alcançar os resultados do estado-da-arte. Em contrapartida, o modelo BERT+Fine-Tunning para o PL-corpus ultrapassou o resultado do estado-da-arte. Neste cenário, para os outros experimentos executados nesta pesquisa, optou-se por utilizar os modelos dos resultados-base (experimentos iniciais) como referenciais para comparação.

Buscando responder à pergunta de pesquisa, foram executados os experimentos como descriminados a seguir:

- Modelo Bi-LSTM+CRF: a) avaliação de modelos de vetores de palavras (*word embbedings*) na etapa de pré-processamento; b) otimização do modelo através dos hiperparâmetros *batch size*, *épocas* e *métodos de otimização*;
- Modelo Transformers (BERT): a) utilização de um modelo pré-treinado para domínio legislativo, com fine-tunning utilizando datasets dos artigos-base; b) otimização do modelo através dos hiperparâmetros: *batch size*, *épocas e taxas de aprendizado.*

O código-fonte do modelo Bi-LSTM+CRF e o corpus utilizado no artigo base estão disponíveis no github dos autores[1]. O modelo BERT foi adaptado de Guilhou[33]. Os corpora dos artigos-base foram divididos entre os conjuntos de treinamento+validação (75%) e teste (25%).

Para a avaliação da otimização dos hiperparâmetros, foram feitas analises isoladas por grupos e através da combinação dos hiperparâmetros (Tabela I). Para cada grupo, foram feitas cinco execuções, modificando aleatoriamente os conjuntos de treinamento, validação e teste, e calculado a média e desvio padrão. Para a etapa de permutação de hiperparâmetros, os corpora de treinamento e teste foram randomizados uma única vez. Os experimentos foram executados na plataforma do Google Colab, utilizando a infraestrutura de hardware GPU NVidia-SMI 460.32.03, com 27.3 GB de memória RAM. Os códigos e os corpora utilizados estão disponíveis na *github* desta pesquisa[6].

A. *Avaliação de Vetores de Palavras (Word Embbedings).*

Vetores de Palavras são representações numéricas de cada palavra (*tokens*) do corpus e o seu grau de similaridade com as demais palavras do corpus, sendo frequentemente utilizados em tarefas de NLP [34]. Os vetores de palavras utilizados foram *Glove* (matriz de tokens, onde cada elemento é a probabilidade de outro token), *Word2Vec* (utilizando a técnica de *continuous bag-of-words,* tenta predizer tokens a partir de uma sequência de palavras), *FastText* (utilizando vetores de palavras como *n*-grams e a soma das representações de palavras) e *Wang2Vec* (variação do *Word2Vec*, que busca suprir a falta de ordem das palavras). As versões destes modelos utilizadas foram pré-treinadas para o português[7]. O código-base do modelo Bi-LSTM+CRF utiliza o vetor de palavras *Glove*, treinado para o português brasileiro. No experimento, buscamos avaliar o impacto destes vetores na etapa de pré-processamento do texto. Em todos os casos, foram aplicados os modelos com 300 dimensões. Os resultados deste experimento são apresentados na Figura 01. O modelo BERT utilizado não utilizou modelos de *word embbedings* e, por isso, não foi utilizado no experimento.

B. *Avaliação de Batch Sizes..*

O tamanho do lote (*Batch Size*) é utilizado para definir o número de exemplos de treinamento em cada iteração, devendo ser ajustado considerando o espaço de memória disponível. Buscou-se investigar sua dependência e o F1-score. As Figuras 2(a-b) e 3(a-b) apresentam os resultados dos experimentos para modelos Bi-LSTM+CRF e BERT, respectivamente.

C. *Avaliação de Número de Épocas.*

O número de épocas define a quantidade de iterações que o algoritmo irá executar nas operações de treinamento. Uma vez que o número de épocas pode influenciar na adaptação do modelo ao corpus analisado, a escolha adequada do número de épocas deve ser feita com prudência, evitando *underfitting* e *overfitting*. Buscamos identificar neste experimento o conjunto de épocas definidos no espaço de configurações arbitrariamente definidos, e sua influência no resultado final do F1-score. As Figuras 2(c-d) e 3(c-d) apresentam os resultados.

D. *Avaliação de Métodos de Otimização*

Dentre os parâmetros que podem afetar o desempenho dos modelos, a escolha do otimizador adequado pode reduzir o erro entre os resultados obtidos versus resultados esperados. Buscamos analisar nesta etapada do experimento influência da escolha do otimizador no resultado final. Entre os métodos utilizados na academia, foram escolhidos: SGD (calcula o gradiente dos parâmetros usando apenas um ou alguns exemplos de treinamento), ADAM (recalcula taxas de aprendizagem adaptativa para cada parâmetro), RMSprop (divide a taxa de aprendizagem por uma média de gradientes quadrados exponencialmente descendente) e ADAGRAD (que adapta a taxa de aprendizado aos parâmetros, executando atualizações maiores para atualizações pouco frequentes e menores para parâmetros frequentes). O modelo Bi-LSTM+CRF apresentado pelo artigo-base utiliza o otimizador SGD como padrão. Buscamos identificar a influência dos outros otimizadores no resultado final. Vale uma observação: não foi possível observar no modelo BERT utilizado um parâmetro associado que possibilitasse a verificação de outros métodos de otimização e, por isso, este parâmetro não foi utilizado nos experimentos para o BERT. A Figuras 2(e-f) demonstra os resultados obtidos.

E. *Avaliação das Taxas de Aprendizado*

A taxa de aprendizado define a velocidade que os modelos atualizam seus parâmetros no aprendizado, ajustando os pesos em relação à perda de gradiente, podendo influenciar no tempo do processo e na qualidade dos resultados. No experimento utilizando o modelo Bi-LSTM+CRF este parâmetro não foi alterado, por terem sido avaliados métodos de otimização que utilizavam taxas de aprendizado variado, sendo utilizado somente no modelo BERT. A Figuras 3(e-f) apresentam os resultados.

---

[6] *https://github.com/bergoliveira/disciplinaDL*

[7] *http://nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc*

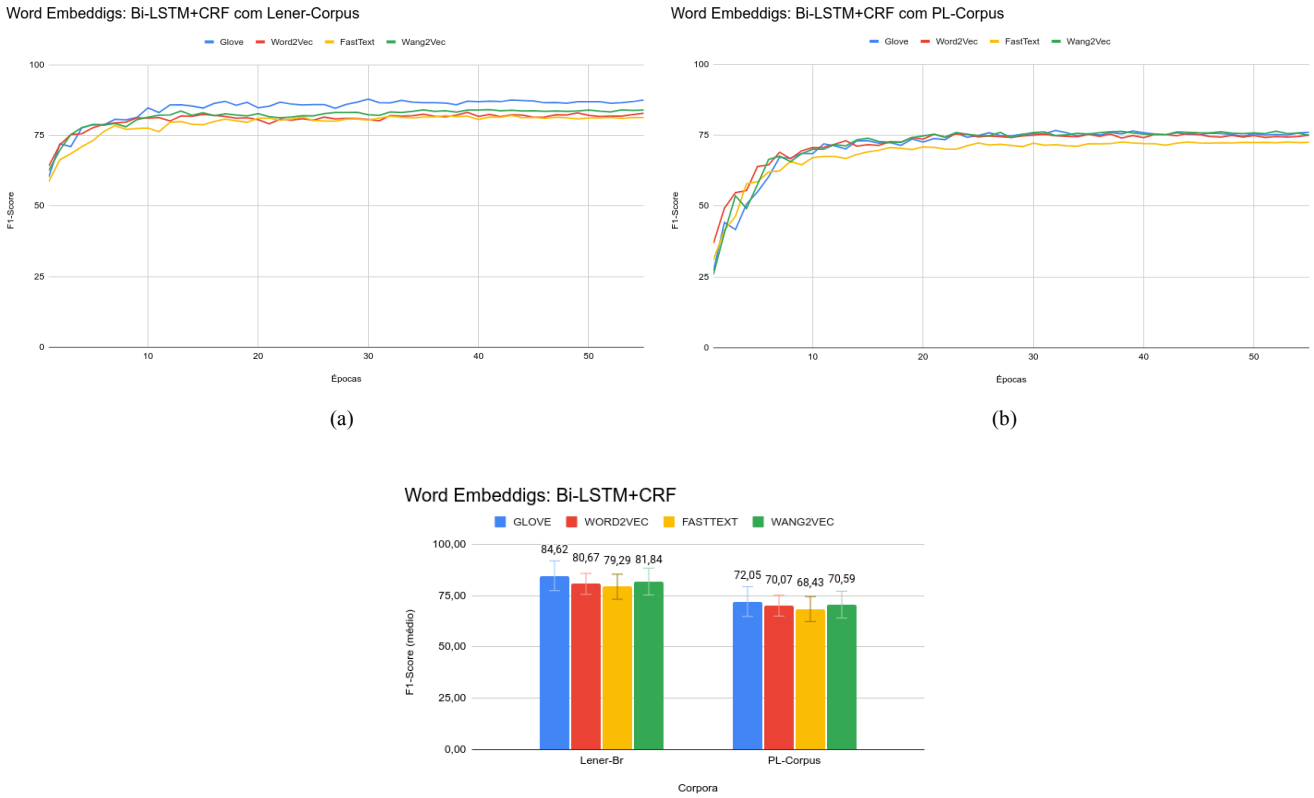

Fig. 1. Análise de *Word Embbedings* no treinamento do modelo Bi-LSTM+CRF: (a) aplicando modelo no Lener-Br corpus, o melhor resultado foi obtido com o *Glove*, alcançando o platô após 30 épocas. Os demais modelos de vetores se mantiveram na média. (b) aplicando o mesmo modelo no PL-Corpus, o vetor FastText apresentou pior desempenho, antes da execução da 10a época, se mantendo abaixo da média dos demais, que apresentaram score médio. (c) Comparando os F1-scores médios após as execuções, o modelo Glove obteve resultado mais satisfatório, considerando média e desvio padrão, em ambos os corpora.

V. Resultados e Discussões

A. *Bi-LSTM+CRF*

Como demonstrado na Fig. 1, o modelo de *word embbedings* que obteve melhor resultado foi o *Glove,* quando aplicado em ambos os corpora. Diante disto, foi escolhido este vetor de palavras como padrão para os demais testes neste modelo. Destacamos um comportamento comum a todos os vetores aplicados: ao chegar numa época em fora alcançado o platô de melhor F1-score (a partir da época 30 em diante, em ambos os corpora), este platô médio se manteve. À exceção a este comportamento foram os vetores FastText e Wang2Vec, que só vieram alcançar este patamar no final do treinamento do PL-Corpus.

Analisando os grupos de hiperparâmetros, para cada grupo testado foram mantidas as demais configurações originais do modelo. Buscou-se investigar inicialmente o efeito do tamanho do batch na etapa de treinamento e o intervalo de generalização. Por questões de infraestrutura de hardware, foi iniciado o experimento com o mesmo valor do artigo-base, dobrando seus valores a cada teste. Como demonstrado na Fig. 2(a-b), o aumento do tamanho batch para o mesmo número de épocas não trouxe benefícios para o treinamento, mas um decréscimo médio na precisão na tarefa de NER, em ambos os corpora.

Analisando o impacto no número de épocas nos conjuntos de treinamento, é possível observar uma pequena tendência de melhoramento no treinamento, principalmente quando aplicado o modelo ao PL-Corpus (Fig. 2c-d), o que indicaria que uma maior número de testes poderia melhorar os resultados encontrados.

Ainda utilizando os mesmos hiperparâmetros e modificando os métodos de otimização, nesta etapa do experimento foi utilizado o modelo original inicialmente (SGD), alterando posteriormente para outros otimizadores. Nos testes utilizados, destaca-se um leve ganho dos otimizadores ADAM e RMSProp na tarefas de NER para o Lener-Br corpus. Para o PL-corpus, apresentou aumento significativo, destacando o desempenho muito baixo do ADAGRAD em todos os casos (vide Fig. 2e-f).

Por fim, buscando encontrar uma melhor configuração para as tarefas de NER nos corpora utilizados, foi executada a permutação de todas as possibilidades do espaço de configurações, incluindo na análise as já executadas anteriormente por grupos. Assim, como demonstra a Fig. 2(g) e Fig. 2h, houve uma coincidente sincronia no treinamento, encontrando o melhor conjunto de hiperparâmetros, para ambos os corpora treinados, com o seguinte espaço de configuração: **[Batchs: 20; Épocas: 75; Otimizador: RMSProp]**. Destacamos um aumento de F1-score do Resultado-Base para o resultado encontrado nestes

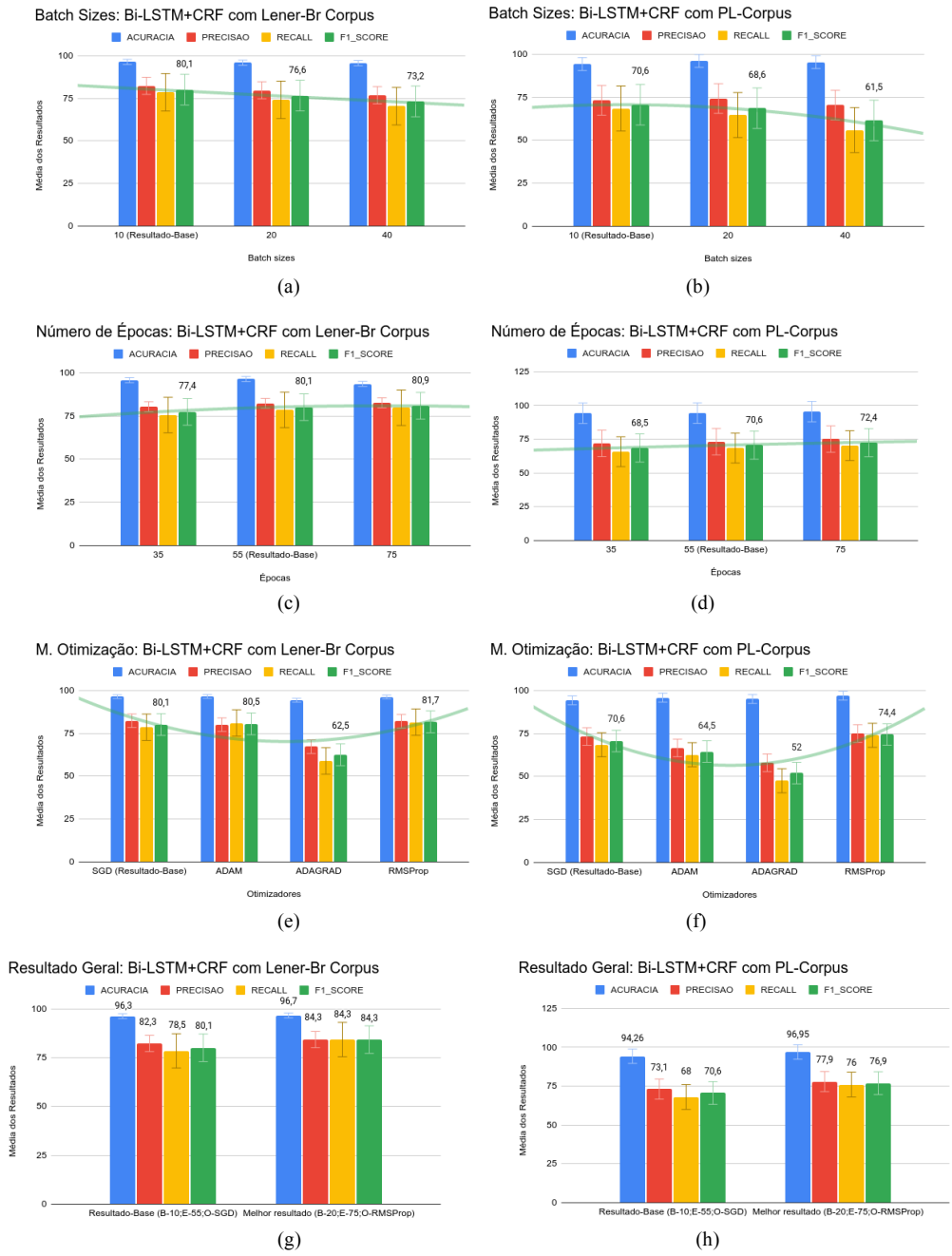

Fig. 2. Comparativo de resultados de experimentos por grupos, utilizando o modelo Bi-LSTM+CRF com glove: (a) e (b) não há influência positiva aumentando o tamanho dos batchs em ambos os corpora; (c) e (d): maior número de épocas demonstrou tendência de melhoramento nas tarefas de NER; (e) e (f): método de otimização com melhor resultado foi RMSProp, principalmente para o PL-Corpus; (g) e (h) o melhor espaço de configuração encontrado para ambos os corpora: [Batchs: 20; Épocas: 75; Otimizador: RMSProp]. Os resultados de comparação destes experimentos os artigos base estão na Tabela V.

experimentos, de 4,2% e 6,3% nos corpora Lener-Br e PL-Corpus, respectivamente. Observamos mediante os dados apresentados anteriormente que a probabilidade de quantidade de épocas associado ao otimizador gradiente RMSProp terem sido os causadores deste ganho.

B. BERT+Fine-Tunning

Buscando avaliar os modelos pré-treinados com BERT+ Fine-Tunning para o domínio legal, foi utilizada a mesma metodologia executada no modelo anterior. Como o modelo BERT não utiliza uma etapa de pré-processamento de texto, esta etapa não foi analisada. Como dito anteriormente, foi treinado um modelo utilizando os parâmetros de batchs e épocas iguais ao Bi-LSTM+CRF, e iniciando a taxa de aprendizado = 2e-5, como arbitrado no modelo pré-treinado de origem.

Iniciando a análise dos resultados pelo tamanho do batch há uma leve tendência de aumento médio de F1-score para o Lener-Br corpus (Fig.3a) à medida que o tamanho aumentava. Para o PL-corpus, por outro lado, este aumento

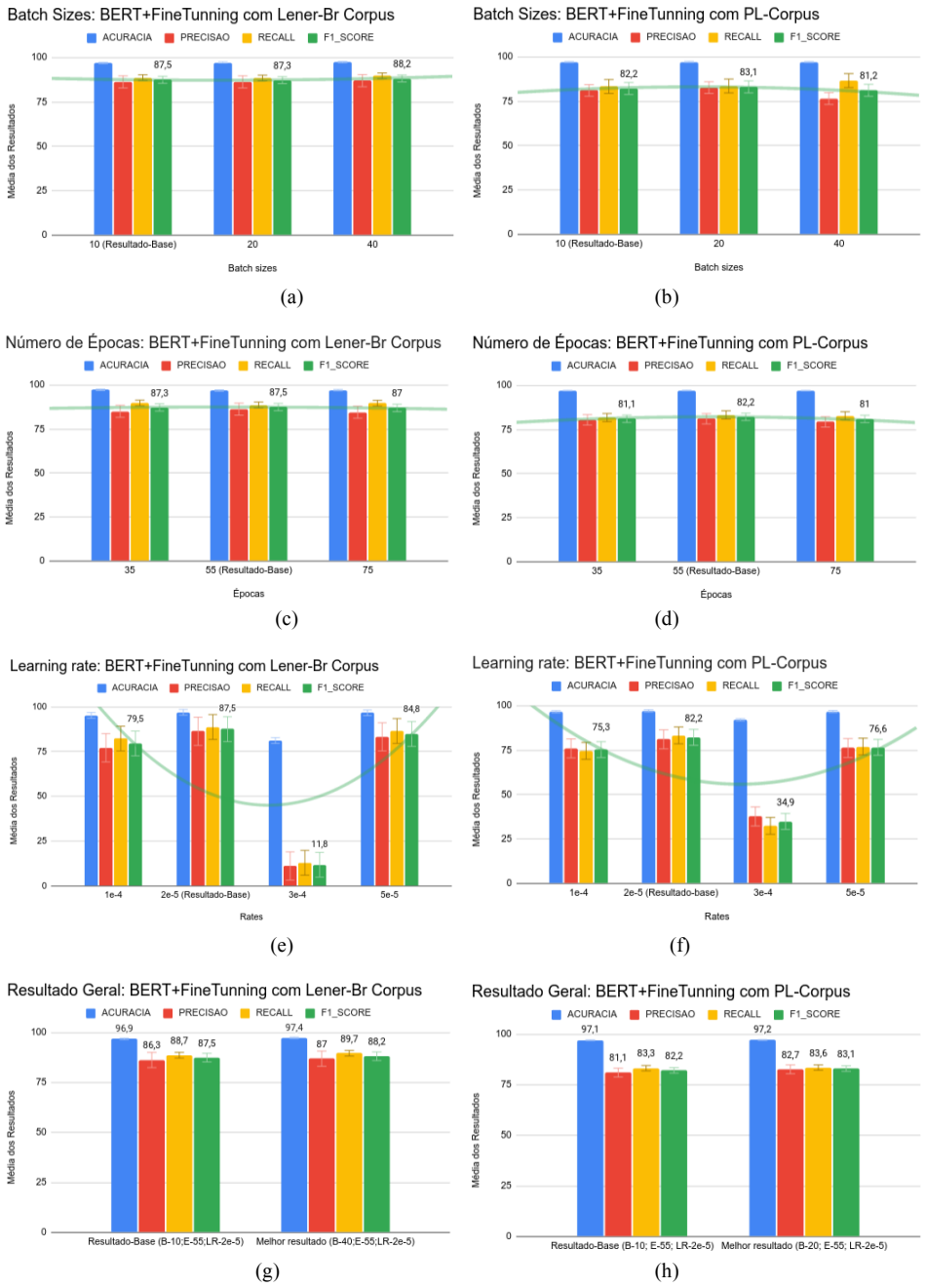

Fig.3 Comparativo de resultados de experimentos por grupos, utilizando o modelo BERT+Fine-Tunning: (a) leve tendência de aumento para o Lener-Br à medida que o tamanho dos batchs aumentava, (b) enquanto que no PL-Corpus, foi o inverso. Todavia, o custo de espaço em disco deve ser considerado. (c) e (d): não houve tendência de crescimento relacionado ao número de épocas, comparados com o artigo-base; (e) e (f): pior desempenho na taxa de aprendizagem com o índice 3e-4, demais índices demonstraram não apresentar relevância comparados com o artigo-base; o melhor espaço de configuração encontrado: (g) para o Lener-Br corpus: [Batchs: 40; Épocas: 55; Taxa de aprendizagem: 2e-5], (h) para o Pl-Corpus: [Batchs: 20; Épocas: 55; Taxa de aprendizagem: 2e-5]. Os resultados de comparação destes experimentos com os artigos base estão na Tabela V

ocorreu quando o batch tamanho 20 foi testado, decrescendo com o aumento seguinte (Fig.3b). Importante ressaltar que, à medida que se aumentava o tamanho dos batchs, a etapa de treinamento utilizava um considerável espaço em disco, o que, na nossa avaliação, não justificaria o seu uso. Na análise do aumento do número de épocas (Fig. 3c-d), houve um decréscimo nas tarefas de NER para ambos os corpora aplicados, comparados com o número de épocas do artigo-base, permanecendo como a melhor configuração. Não encontramos até o momento respostas para este resultado.

Quanto à taxa de aprendizagem (Fig. 3e-f), o índice = 3e-4 apresentou baixíssima qualidade de aprendizado em ambos os corpora, chegando, em alguns momentos, a não identificar nenhuma entidade no treinamento. Dentre as outras configurações analisadas, o índice utilizado no artigo-base se manteve como a melhor opção.

Finalmente, após a combinação de todos os hiperparâmetros do espaço de configurações para este modelo, utilizados no conjunto de treinamento em ambos os

TABLE IV.
COMPARAÇÃO DE RESULTADOS ENTRE EXPERIMENTOS[1]

Modelo	Corpus	Entidades	Resultados[2]							
			Resultado-base				Melhor resultado dos experimentos			
			AC	PR	RC	F1[3]	AC	PR	RC	F1[3]
Bi-LSTM+CRF com Glove	Lener-Br	Pessoa	---	94,1±5,3	93,6±8,1	93,8±7,0	---	96,3±8.4	98,7±7	**97,5±7,8**
		Local	---	45,4±12,2	55,8±14,3	49,3±11,6	---	56,5±12.8	69,8±15,2	**61,2±13,15**
		Organização	---	83,5±6,6	74,0±14,1	78,0±12	--	82,5±8,9	77,8±11.7	**79,94±10,5**
		Tempo	---	86,5±1,8	88,5±3	87,4±1,8	---	87,3±4,2	89,6±1,0	**88,4±2,5**
		Legislação	---	86,4±4,6	85,8±8,8	86,0±7,2	---	84,0±3,5	88,4±2,6	**86,15±2,9**
		Jurisprudência	---	68,4±10,8	64,9±12,8	66,2±12	---	88,07±4,2	85,0±10,6	**86,27±8,1**
		Overall[6]	96,3±1,5	82,3±3,3	78,5±10,1	80,1±7,8	96,7±1,0	84,3±5,1	84,3±7,4	**84,3±6,2**
	PL-Corpus	Pessoa	--	78,6±2.8	67.9±7.8	72.6±5.7	---	84,4±3,1	74,8±3,5	**79,3±3,1**
		Local	--	74,2±3.7	74.3±16.9	72.9±14.4	---	78,8±3,0	80,5±8,3	**79,5±6,4**
		Organização	--	70,1±10.2	54.1±16.8	60.1±16.4	---	73,8±3,4	72,3±6,6	**73,0±5,1**
		Data	--	87.1±3.2	90.7±3.6	88.8±3.31	---	87,8±1,4	91,2±0,9	**89,5±0,9**
		Evento	--	80.9±36.9	31.1±14	44.8±20.1	---	91,8±20,4	34,5±7,6	**49,9±10,9**
		Fundamento	--	76.4±7.7	75.6±8.5	75.9±8.1	---	80,2±3,2	78,7±4,4	**79,4±3,6**
		Produto de Lei	--	41.0±16.4	26.8±9.6	32.2±12.3	---	43,4±8,0	42,2±6,3	**42,5±6,5**
		Overall	94,2±8,8	73,1±10,5	68,4±11,0	70,6±10.7	96,9±0,4	77,9±2,3	76,0±4,9	**76,9±3,8**
BERT+Fine Tunning	Lener-Br	Pessoa	---	99,4±4,8	98,3±7,6	**98,8±6,5**	---	98,6±2,2	97,3±1,3	97,91,7
		Local	---	76,7±11,7	86,6±13,9	81,3±11,1	---	79,4±4,5	87,6±14,5	**83,3±9,1**
		Organização	---	89,1±6,1	89,0±13,6	89,1±11,5	---	89,1±3,5	88,9±2,4	**89±2,8**
		Tempo	---	94,4±1,2	98,5±2,5	96,9±1,2	---	97±1,1	98,3±0,5	**97,6±0,8**
		Legislação	---	88,0±3,4	83,7±7,5	85,8±6	---	87±3,6	85,5±3,1	**86,2±3,3**
		Jurisprudência	---	76,3±10,1	73,5±12,1	74,8±11,3	---	79,9±6,6	75,6±7,7	**77,7±7**
		Overall[6]	96,9±0,3	86,3±2,8	88,7±1,5	87,5±1,6	97,4±0,4	87±4,7	89,7±1,3	**88,2±2,6**
	PL-Corpus	Pessoa	--	91,5±1	91,1±1,8	91,3±0,8	---	92,3±1,9	90,8±5,7	**91,5±3,6**
		Local	--	80,8±1,4	83,8±2,3	82,3±0,5	---	83,2±2,8	85,1±2,9	**84,1±2,5**
		Organização	--	89,9±3	78,8±3,4	81,7±3,1	---	81,8±5,4	81,8±6,41	**81,8±5,7**
		Data	--	84,9±5,9	96,8±2,1	90,4±3,0	---	87,9±5,4	97,5±3,2	**92,4±3,2**
		Evento	--	100±18,9	62,5±17,5	76,9±18	---	100±13,	62,5±5,5	**76,9±7,2**
		Fundamento	--	77,2±2,3	81,9±5,2	**79,5±3,2**	---	76,2±3,8	81,5±3,2	78,7±3,1
		Produto de Lei	--	49,4±2,5	54,4±3,7	51,8±4,0	---	55,2±5,3	50,6±10,5	**52,8±4,3**
		Overall	97,1±0,3	81,1±2,8	83,3±1,5	82,2±1,6	97,2±0,11	82,7±1,6	83,6±1,11	**83,1±1,1**

[1]Métricas: AC: Acurácia, PR: Precisão, RC: Recalll e F1: F1-score. Os resultados estão em percentuais (%).[2] Os resultados dos experimentos apresentaram melhor performance em quase todos os F1-score, comparados com os Resultados-Base. [3]F1-score foi o fator determinando para decisão de melhor performance. Resultados gerados pelo modelo pré-treinado BERT+Fine-Tunning apresentaram maior média de F1-score e menores valores de desvio-padrão.

TABLE V.
ARTIGO-BASE VERSUS MELHOR RESULTADO[1]

Corpus	Resultados							
	Artigo-base (Overall)[2] (Bi-LSTM+CRF com Glove)[3]				Melhor resultado nos experimentos (BERT+Fine-Tunning)[4]			
	AC	PR	RC	F1	AC	PR	RC	F1
Lener-Br	(ni)	93,2	91,9	**92,5**	97,4±0,4	87±4,7	89,7±1,3	88,2±2,6
PL-Corpus	97,2±0,7	83,4±0,9	70,4±1,5	76,2±1,1	97,2±0,11	82,7±1,6	83,6±1,11	**83,1±1,1**

[1]Métricas: AC: Acurácia, PR: Precisão, RC: Recalll e F1: F1-score. Os resultados estão em percentuais (%) [2]Resultados do Artigo-base não apresentam média e desvio-padrão. [3]Resultados do Artigo-Base possuem melhor performance que os experimentos executados. [4]Resultados obtidos pelo BERT+Fine-Tunning foram bem mais relevantes que os resultados do Artigo-Base.

corpora, as melhores configurações encontradas foram: (i) para ao Lener-Br corpus: **[Batchs: 40; Épocas: 55; Taxa de aprendizagem: 2e-5]**, com um pequeno aumento de 0,7% de F1-score (vide Fig. 3g), devendo ser considerado a utilização de espaço em disco, como dito anteriormente; (ii) para o PL-Corpus: **[Batchs: 20; Épocas: 55; Taxa de aprendizagem: 2e-5]**, com o aumento de 0,9% de F1-score. Em ambos os casos, o parâmetro comparação foi o Resultado-Base, apresentado anteriormente na Tabela II.

C. *Análise geral*

Fazendo um comparativo entre os resultados obtidos nos experimentos, e comparando estes resultados com os Resultados-Base apontados na Tabela IV, é possível perceber: (i) os resultados encontrados nos experimentos aplicados obtiveram melhor performance que os Resultados Base; (ii) os dados gerados pelo modelo BERT+Fine-Tunning tiveram maior precisão na média do F1-score e menor desvio-padrão, demonstrando melhor qualidade nas tarefas de NER; (iii) para ambos os corpora, o modelo BERT+Fine-Tunning obteve o melhor resultado de F1-score geral, obtendo 88,2±2,6% para o Lener-Br corpus e 83,1±1,1% para o PL-corpus.

Finalmente, comparando os melhores resultados dos experimentos com os resultados dos artigos-base, é possível perceber que os resultados apresentados pelos autores do

Lener-Br [19] ainda se mantém superiores. Por outro lado, os resultados obtidos para o PL-Corpus nos experimentos executados nesta pesquisa apresentaram 9% de ganho de F1-score médio nos resultados comparados com o artigo-base [22]. Os resultados encontrados demonstram que são necessárias mais investigações para descobrir como alcançar os resultados obtidos no artigo-base para ao Lener-Br corpus, e como melhorar os resultados das tarefas de NER do PL-Corpus para o mesmo patamar.

VI. Conclusão

Este trabalho é resultado de uma pesquisa exploratória no campo de Reconhecimento de Entidades Nomeadas (NER) no domínio jurídico em língua portuguesa, utilizando Deep Learning. Utilizando dois modelos de aprendizado encontrados na literatura para o domínio estudado, Bi-LSTM+CRF com Glove e BERT+Fine-Tunning, a pesquisa buscou inicialmente reproduzir o estado-da-arte e, a partir da manipulação do espaço de configurações destes modelos, apontar o melhor desempenho encontrado em termos de Acurácia, Precisão, Recall e F1-score. Foram utilizados dois corpora públicos do domínio legal e anotados no padrão CoNLL-2002, os resultados experimentais conseguiram demonstrar que o modelo BERT_Fine-Tunning conseguiu resultados superiores ao outro modelo, com F1-scores 88,2±2,6% para o Lener-Br corpus e 83,1±1,1% para o PL-corpus, um aumento de aproximadamente 9% de ganho de F1-score médio neste último corpus. Além disto, em trabalhos futuros, pretende-se investigar melhor configuração para alcançar os resultados do artigo-base pioneiro, além de investigar outros métodos de avalição, como a influencia de data augmentation e loss nas tarefas de NER.

References

[1] M. B. Finatto, L. Lopes, and L. Silva. “Processamento de Linguagem Natural, Linguística de Corpus e Estudos Linguísticos: uma parceria bem-sucedida”. In: Revista Domínios de lingu@gem. Uberlândia, MG. Vol. 9, n. 5, 2015.

[2] D. Maynard, K. Bontcheva, I. Augenstein. “Natural language processing for the semantic web”. Synthesis Lectures on the Semantic Web: Theory and Technology, 2016.

[3] S., Collovini, T.L. Bonamigo, R. Vieira. “A review on relation extraction with an eye on portuguese”. In: Journal of the Brazilian Computer Society, 19, 2013.

[4] A. Akbik, L., Chiticariu, M. Danilevsky, Y. Kbrom, Y. Li, H. Zhu. “Multilingual Information Extraction with PolyglotIE”. In: Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: System Demonstrations, 2016.

[5] D.O.F. do Amaral, R. Vieira. “NERP-CRF: uma ferramenta para o reconhecimento de entidades nomeadas por meio de Conditional Random Fields”. Linguamática, 2014.

[6] J.P.C. Pirovani. “CRF+LG: uma abordagem híbrida para o reconhecimento de entidades nomeadas em português”. PhD thesis. Universidade Federal do Espírito Santo, 2019.

[7] F. de Grove, K. Boghe, L. de Marez. “(What) Can Journalism Studies Learn from Supervised Machine Learning?”. In: Journalism Studies, 2020.

[8] T.H. Dang, H.-Q. Le, T.M. Nguyen, S.T. Vu. “D3NER: biomedical named entity recognition using CRF-BILSTM improved with fine-tuned embeddings of various linguistic information”. Bioinformatics, 2018.

[9] P.H. Luz de Araujo, T.T. de Campos, R.R.R. de Oliveira, M. Stauffer, S. Couto, 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), 2018

[10] D.S.C. Pedroso, M. Ladeira, T.P. Faleiros. “Does Semantic Search Performs Better than Lexical Search in the Task of Assisting Legal Opinion Writing?”. In: 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019.

[11] Z. Hong, Q. Zhou, R. Zhang, W. Li, T. Mo. “Legal Feature Enhanced Semantic Matching Network for Similar Case Matching”. In: International Joint Conference on Neural Networks (IJCNN), 2020.

[12] E. Leitner, G. Rehm, J. Moreno-Schneider. “Fine-Grained Named Entity Recognition in Legal Documents”. In: Semantic Systems. The Power of AI and Knowledge Graphs. SEMANTiCS 2019. Lecture Notes in Computer Science, vol 11702. Springer, Cham, 2020.

[13] E. Leitner, G. Rehm, J. Moreno-Schneider. “A dataset of German legal documents for named entity recognition”. In: LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings, 2020.

[14] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, T. Zhang, X. Han, Z. Hu, H. Wang, J. Xu.“CAIL2019-SCM: A Dataset of Similar Case Matching in Legal Domain”. ARXiv API: 1911.08962, 2019.

[15] P. Bhattacharya, K., Ghosh, A. Pal, S. Ghosh. “Methods for Computing Legal Document Similarity: A Comparative Study”. ARXiv API: 2004.12307, 2020.

[16] P.V. Quinta de Castro. “Aprendizagem Profunda para Reconhecimento de Entidades Nomeadas em Domínio Jurídico”. Dissertação de Mestrado. Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, 2019.

[17] R. Socher and C. Manning. “Deep Learning for NLP”. HLT-NAACL Tutorials, 2013.

[18] R. Dale. “Law and word order: Nlp in legal tech”. Natural Language Engineering, 25(1), 2019.

[19] J. Bergstra, et al. “Algorithms for hyper-parameter optimization”. In: Advances in Neural Information Processing Systems 24 (NIPS 2011), 2011.

[20] Y. Zhang and B. Wallace. “A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classication”. arXiv preprint arXiv:1510.03820 , 2015.

[21] L. Tuggener, M. Amirian, F. Benites, P. von Däniken, P. Gupta, F.-P. Schilling, T. Stadelmann. “Design Patterns for Resource-Constrained Automated Deep-Learning Methods”. https://doi.org/10.3390/ai1040031, 2020.

[22] H.O. Albuquerque, R. Costa, G. Silvestre, E. Souza, N.F.F. Silva, D. Vitório, G. Moriyama, L. Martins, L. Soezima, A. Nunes, F. Siqueira, J.P. Tarrega, J.V. Beinotti, M. Dias, M. Silva, M. Gardini, V. Silva, A.C.P. L.F. Carvalho, A.L.I. Oliveira. “UlyssesNER-Br: a corpus of Brazilian legislative documents for named entity recognition”. In:International Conference on the Computational Processing of Portuguese (PROPOR). In Press, 2022.

[23] P. Quaresma, T. Gonçalves. "Using Linguistic Information and Machine Learning Techniques to Identify Entities from Juridical Documents". Springer, 2010.

[24] I. Angelidis, I. Chalkidis, M. Koubarakis. "Named entity recognition, linking and generation for Greek legislation". In: Proceedings of 31st International Conference on Legal Knowledge and Information Systems (JURIX 2018), 2018.

[25] I. Badji. "Legal entity extraction with NER systems". Master’s thesis, Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid, 2018.

[26] T. Váradi et al. "The MARCELL Legislative Corpus". In: Proceedings of the 12th Language Resources and Evaluation Conference, European Language Resources Association, 2020.

[27] D. Santos, N. Cardoso. “A golden resource for named entity recognition in Portuguese”. In: International Conference on Computational Processing of the Portuguese Language, PROPOR 2006, 2006.

[28] V.J. Alles. “Construção de um corpus para extrair entidades nomeadas do Diário Oficial da União utilizando aprendizado supervisionado”. Master’s thesis, Departamento de Engenharia Elétrica, Universidade de Brasília, 2018.

[29] P.H. Luz de Araujo, T.E. Campos, F.A. Braz, N.C. Silva. “VICTOR: a dataset for Brazilian legal documents classification”. In: Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), 2020.

[30] Z. Wang, Y. Wu, P. Lei, C. Peng. “Named Entity Recognition Method of Brazilian Legal Text based on pre-training model”. In: Journal of Physics: Conference Series, 2020.

[31] A. Aguiar, R. Silveira, V. Pinheiro, V. Furtado, J.A. Neto. “Text Classification in Legal Documents Extracted from Lawsuits in Brazilian Courts”. In: 10th Brazilian Conference, BRACIS 2021, 2021.

[32] E.F.T.K. Sang. “Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition”. In: COLING-02: The 6th Conference on Natural Language Learning, 2002.

[33] P. Guillou. “(BERT base) NER model in the legal domain in Portuguese (LeNER-Br)”. Disponível em https://huggingface.co/pierreguillou/ner-bert-base-cased-pt-lenerbr, 2021.

[34] N. Hartmann, E. Fonseca, C. Shulby, M. Treviso, J. Rodrigues, S. Aluisio. “Portuguese word embeddings: Evaluating on word analogies and natural language tasks”. arXiv preprint arXiv:1708.06025 , 2017.