Avaliação *In-Domain* e *Cross-Domain* em Restauração de Pontuação Utilizando Processamento de Linguagem Natural

Brenda C. D. Moura
brendacdmoura@gmail.com
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - *Campus* Manaus Zona Leste
Manaus, Amazonas, Brasil

Angel G. de S. Sales
angelgabrieldesouzasales@gmail.com
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - *Campus* Manaus Zona Leste
Manaus, Amazonas, Brasil

José E. B. de S. Linhares
breno.linhares@ifam.edu.br
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - *Campus* Manaus Zona Leste
Manaus, Amazonas, Brasil

Fabiann M. D. Barbosa
fabiann.dantas@ifam.edu.br
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - *Campus* Manaus Zona Leste
Manaus, Amazonas, Brasil

Amadeu A. Neto
amadeu.neto@ifam.edu.br
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - *Campus* Manaus Zona Leste
Manaus, Amazonas, Brasil

ABSTRACT

Punctuation plays a fundamental role in conveying the correct meaning in written texts. As a result, punctuation errors can occur, significantly impairing the way a message is interpreted, whether in formal or informal contexts. In this sense, the use of machine learning, combined with recent techniques in natural language processing, has been widely used in the task of punctuation restoration, in languages such as English. However, despite the wide application of this task in other languages, its use in Portuguese is still quite limited. In this work, we propose to adapt a punctuation restoration model for its application in formal texts in the Portuguese language, in addition to evaluating the model's behavior in informal texts. The Portuguese Legal Sentences v3 dataset was used to train the model, which was also used for the in-domain evaluation. Regarding the cross-domain evaluation, the IWSLT (International Workshop on Spoken Language Translation) database was used, consisting of transcripts of lectures known as TED Talks. The results indicate that the model with the largest amount of training data and that mapped all question marks to full stops performed satisfactorily in the formal context, suggesting that the methodology adopted was adequate for the proposed task. Furthermore, it was found that the scarcity of question marks negatively impacts the model's performance and that, in the informal context, the results were unsatisfactory in the evaluation metrics, suggesting that formal and informal sentences have their own structures, which the model was unable to generalize adequately in the informal context.

KEYWORDS

Punctuation marks, Recurrent neural network, LTSM, Natural language processing, Punctuation restoration

1 INTRODUÇÃO

A escrita desempenha um papel fundamental na vida das pessoas, seja no âmbito pessoal, como na comunicação via aplicativos de mensagens, seja no contexto acadêmico, em provas de redação, como o ENEM, utilizado no processo seletivo de universidades e institutos federais. Além disso, no ambiente profissional, a escrita é essencial na elaboração de relatórios e outras atividades que exigem precisão e clareza. Assim, a importância da escrita, tanto formal quanto informal, é evidente em diversas áreas. A aplicação correta das regras de pontuação é fundamental, pois o uso inadequado pode comprometer a compreensão do texto [1].

A pontuação é um recurso da ortografia que pode ser definida como um sistema de sinais gráficos, composta, por exemplo, por ponto, vírgula e ponto de interrogação. A função da pontuação é indicar a estrutura e a organização de uma frase ou texto [2]. Dessa forma, a ausência ou o uso inadequado desses sinais gráficos pode comprometer a correta interpretação de um texto. Ramos *et al.* (2022) realizaram um estudo sobre a ocorrência de quatro categorias de erros nas redações do Exame Nacional do Ensino Médio (ENEM) de 2019. As categorias analisadas foram: pontuação, concordância, ortografia e problemas estruturais. Como resultado, verificou-se que os erros de pontuação ocorrem com mais frequência do que erros das demais categorias, prejudicando a clareza dos textos [3].

A partir da necessidade de aprimorar o conhecimento de estudantes e profissionais sobre pontuação, buscou-se formas de integrar o Aprendizado de Máquina (AM) às correções automáticas de pontuação. É importante ressaltar que é possível realizar a correção manual. No entanto, essa abordagem possui limitações. No caso de textos longos ou após múltiplas revisões, erros podem passar despercebidos, devido ao cansaço. Portanto, torna-se recomendável a utilização de um sistema automatizado para correções (ou restaurações) desses sinais gráficos. Diante desse cenário, uma subárea do AM, voltada para dados textuais, chamada de Processamento de Linguagem Natural (PLN), vem ganhando destaque, por contribuir para essa tarefa de restauração automática de pontuação. Neste trabalho, propõe-se explorar o estado da arte em AM no contexto da restauração de pontuação, com o objetivo de desenvolver um modelo de aprendizado capaz de identificar e aplicar corretamente os sinais de pontuação em textos formais no idioma português[1]. Adicionalmente, para compreender o desempenho do modelo em

[1]Neste cenário, caracteriza-se uma avaliação *in-domain*, a qual consiste em utilizar para treinamento e testes dados que pertencem ao mesmo domínio.

um cenário de linguagem informal, propõe-se uma avaliação *cross-domain*[2].

Nas seções a seguir, serão apresentados os trabalhos relacionados (Seção 2), a metodologia proposta (Seção 3), os resultados alcançados (Seção 4) e as conclusões (Seção 5), com direções para trabalhos futuros em relação à pesquisa apresentada.

2 TRABALHOS RELACIONADOS

A restauração de pontuação é uma tarefa de grande relevância no campo de PLN, especialmente no pós-processamento de textos gerados por sistemas de reconhecimento de fala. Diversos estudos têm contribuído para o estado da arte, com o objetivo de identificar metodologias de maior eficácia para essa tarefa. Nesta seção, apresentam-se três pesquisas que aplicaram diferentes técnicas e bases de dados para restaurar pontuação em textos no idioma português. A seleção dos trabalhos levou em consideração aqueles mais alinhados ao tema, ou seja, que abordaram a restauração de pontuação em textos em português ou que discutiram avaliações em bases de dados com diferentes estilos. Ademais, optou-se por estudos que refletissem o estado atual da arte, resultando na escolha de trabalhos publicados a partir de 2021.

No trabalho proposto por [4], apresenta-se uma abordagem para a restauração de pontuação e capitalização de textos em espanhol e português. O conjunto de dados utilizado foi o *OpusParaCrawl* [5], composto por dados paralelos extraídos de *websites* multilíngues. A metodologia proposta baseia-se no ajuste fino de diferentes modelos pré-treinados monolíngues, como BETO e BR_BERTo, e no modelo multilíngue XLM-RoBERTa. A avaliação de desempenho foi conduzida com base em métricas ponderadas de precisão, *recall* e *f1-score*, além da média simples do *f1-score*. Os resultados demonstraram que, considerando a média simples do *f1-score*, os modelos BETO e XLM-RoBERTa apresentaram os melhores desempenhos para os textos em espanhol e português, respectivamente.

Na pesquisa conduzida por [6], aplicam-se as arquiteturas *Base* e *Large* dos modelos BERT e T5 para a tarefa de restauração de pontuação em textos educacionais do português brasileiro. Os autores utilizaram a base de dados NILC, composta por textos extraídos de livros didáticos. Além disso, com o objetivo de avaliar a efetividade dos modelos em um ambiente real, foi realizada uma avaliação *cross-domain* utilizando o *dataset* MEC, que é composto por redações de estudantes do ensino médio. Para mensurar o desempenho, foram aplicadas as métricas de precisão, *recall* e *f1-score*. Adicionalmente, para os modelos baseados em T5, foi aplicado o coeficiente BLEU. Os resultados indicaram que a arquitetura *T5 Large* apresentou o melhor desempenho *in-domain*, alcançando 89% de *f1-score* geral. No entanto, na avaliação *cross-domain*, *BERT Base* destacou-se, com 73% na mesma métrica.

No estudo proposto por [7], analisaram-se três modelos de restauração de pontuação em textos no idioma português brasileiro. As arquiteturas avaliadas foram: CRF (referencial), Bi-LSTM + CRF e BERT [8]. Para o treinamento, foi utilizado o conjunto de dados IWSLT [9], composto por transcrições das palestras *TED Talks*. Além disso, para realizar uma avaliação *cross-domain*, foi empregado o *dataset* OBRAS [10], composto por textos literários. Os resultados foram avaliados com base nas métricas de precisão, *recall* e *f1-score*. A partir da média das métricas, concluiu-se que o modelo BERT obteve a melhor perfomance dentre os demais, obtendo 81% e 73.5% de *f1-score* nos cenários *in-domain* e *cross-domain*, respectivamente. Entretanto, uma análise dos custos de recursos de GPU revelou que o modelo BERT exigiu uma quantidade significativa de recursos computacionais para seu treinamento e testes.

A partir da análise dos trabalhos apresentados na Tabela 1, observa-se que a maioria das pesquisas atuais explora o uso de modelos pré-treinados baseados em *transformers* para a restauração de pontuação em português. Embora essa abordagem ofereça resultados satisfatórios, ainda é necessário experimentar soluções que sejam menos custosas computacionalmente, visando alternativas para contextos com recursos tecnológicos limitados. Nesse sentido, este trabalho avalia o desempenho de um modelo Bi-LSTM + CRF + *Self-Attention*. Além disso, realiza-se também uma análise *cross-domain*, em que o desempenho do modelo é avaliado em textos formais e informais, de maneira semelhante ao abordado por [7], porém o treinamento sendo realizado a partir de textos formais.

3 METODOLOGIA PROPOSTA

Nesta seção, apresenta-se a metodologia empregada para a restauração de pontuação, a partir de dados textuais. Conforme o diagrama em blocos ilustrado na Figura 1, o sistema é dividido em duas principais etapas, sendo: i) treinamento e ii) inferência. Para cada etapa, realizou-se o tratamento das bases de dados, aplicando técnicas comuns ao problema de restauração de pontuação. Em seguida, treinou-se o modelo e o avaliou, sob contextos *in-domain* e *cross-domain*. Em relação à arquitetura utilizada no treinamento dos modelos, adotou-se a proposta de [11], que apresenta uma estrutura similar à descrita em [7], porém com a inclusão de um mecanismo de atenção, o que aprimora a capacidade do modelo de capturar dependências contextuais e melhorar a precisão na restauração da pontuação. Cada um dos blocos serão detalhados a seguir.

3.1 Base de dados

Nesta pesquisa, o foco de investigação inicial foram dados textuais, caracterizados pela formalidade (norma culta da língua) e por pertencerem ao idioma português. Para isso, definiu-se uma base de dados composta essencialmente por textos formais chamado *Portuguese Legal Sentences v3* [12], que é composta por sentenças da Suprema Corte de Justiça.

A separação da base de dados selecionada em subconjuntos de treino, validação e testes foi realizada utilizando a biblioteca Python chamada *Sklearn*. O subconjunto de treino foi chamado de *Train*, de validação foi chamado de *Dev* e o de testes foi subdividido em dois, sendo chamados de *Ref* e *ASR*, para adequar-se a estrutura da base de dados empregada na *baseline* adaptada [11], onde ASR advém de *Automatic Speech Recognition* e refere-se a textos gerados automaticamente. No entanto, neste trabalho, a sigla é utilizada apenas para facilitar a aplicação dos testes.

Para analisar a metodologia proposta na tarefa de restauração de pontuação, foram utilizados três modelos de aprendizado, aplicando-os em diferentes cenários. O primeiro cenário consistiu na utilização de 5% da base de dados, considerando as classes: vírgula (*comma*), ponto final (*period*) e ponto de interrogação (*questionmark*). No

[2] *Cross-domain* é um conceito aplicado quando os dados de testes são de um domínio diferente, mas relacionado aos dados de treinamento.

Tabela 1: Comparação entre os trabalhos relacionados e a pesquisa proposta.

Autoria	Arquitetura	Base de Dados	Idioma	Métricas	Avaliação
Pan, Diaz e Valencia-Garcia [4]	XLM-RoBERTa BETO, ALBERTO, DistilBETO MarIA, BERTIN BERTimbau Base, BR_BERT	*OpusParaCrawl*	Espanhol, Português	Precisão, *Recall*, *F1-score*	*In-domain*
Lima *et. al* [6]	BERT (*Base* e *Large*) T5 (*Base* e *Large*)	NILC MEC	Português Brasileiro	Precisão, *Recall*, *F1-score*, BLEU	*In-domain*, *Cross-domain*
Lima *et. al* [7]	CRF Bi-LSTM + CRF BERT	IWLST 2012 OBRAS	Português Brasileiro	Precisão, *Recall*, *F1-score*	*In-domain*, *Cross-domain*
Pesquisa proposta	Bi-LSTM + CRF + *Self attention*	*Portuguese Legal Sentences v3* IWSLT 2014 - 2016	Português	Precisão, *Recall*, *F1-score*	*In-domain*, *Cross-domain*

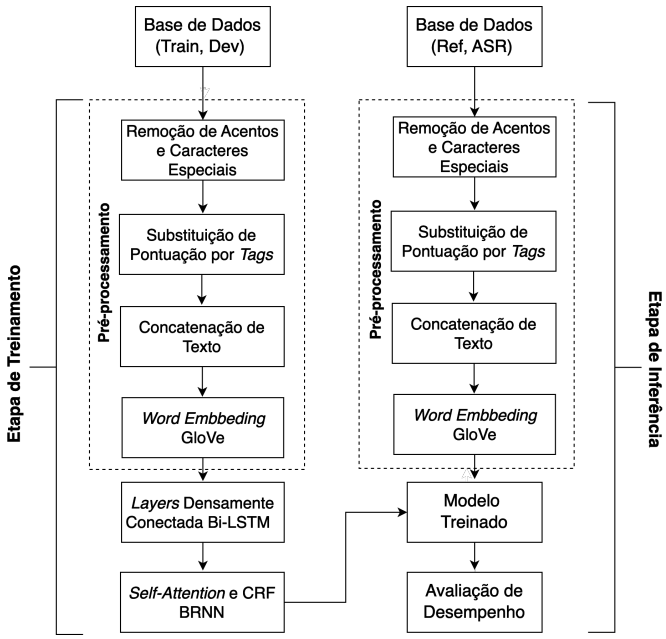

Figura 1: Diagrama em blocos da metodologia proposta.

segundo cenário, também foi utilizado 5% da base de dados, porém substituindo os pontos de interrogação por pontos finais, a fim de verificar se esta alteração melhoraria os resultados do modelo. O terceiro cenário utilizou a estratégia definida para o segundo cenário quanto à substituição de classes, aumentando a porcentagem de utilização da base de dados de 5% para 20%.

As quantidades de dados foram definidas considerando a limitação de recursos computacionais, que impossibilitou o treinamento em larga escala. Além disso, a base de dados selecionada para o treinamento contém um volume significativo de informações, o que reforçou a necessidade de um ajuste no tamanho do conjunto de dados utilizado.

Conforme apresentado na Tabela 2, observa-se que, no primeiro cenário, contém um desbalanceamento de dados entre as classes, devido à baixa quantidade de amostras de pontos de interrogação. Esta situação é inerente em documentos produzidos na área do Direito, por carecer de espaços para questionamentos ou indagações. Por isso, o segundo cenário proposto foi importante para analisar se o desbalanceamento entre as classes prejudicaria o desempenho do modelo.

Adicionalmente, para compreender o desempenho dos modelos a partir dos cenários propostos, realizou-se uma avaliação *cross-domain*, utilizando a base de dados IWSLT (*International Workshop on Spoken Language Translation*), referentes aos anos de 2014-2016. Esta base de dados é composta por transcrições de palestras conhecidas como *TED Talks*, na qual a comunicação é estabelecida, em geral, por vocabulário informal. Por isso, os dados do IWSLT mostraram-se ideais para o propósito desta avaliação.

3.2 Pré-processamento

Para assegurar que a base de dados estava adequada ao método utilizado para restauração de pontuação e minimizar a probabilidade de viés no modelo gerado, foram realizadas quatro etapas de pré-processamento: concatenação de texto, remoção de acentos e caracteres especiais, substituição de pontuação por *tags* e aplicação do *word embedding* GloVe.

Na primeira fase, todas as linhas do subconjunto de dados (*Train*, *Dev*, *Ref* e *ASR*) foram concatenadas, a fim de obter um único arquivo de texto para cada base de dados. Em seguida, foi realizada a remoção de acentos e caracteres especiais. Para esta remoção, utilizou-se o Formato de Normalização de Compatibilidade de Decomposição (NFKD), o qual decompõe os caracteres em seus componentes básicos como, por exemplo, o caractere *à* decomposto em *a* e acento *crase*. Após a normalização das *strings*, os textos foram codificados para o formato ASCII e decodificados de volta para o formato UTF-8, finalizando a segunda etapa do pré-processamento.

A próxima fase consistiu na substituição dos sinais de pontuação pelas *tags* <comma>, <period> ou <questionmark>, as quais representam vírgula, ponto final e ponto de interrogação, respectivamente. É importante destacar que, nos dois últimos cenários propostos, todos os pontos de interrogação foram mapeados para a *tag* <period>, uma modificação comumente utilizada em pesquisas relacionadas à restauração de pontuação, devido à similaridade de uso entre o ponto de interrogação e o ponto final.

Tabela 2: Detalhes dos conjuntos de dados utilizados nos cenários 1, 2 e 3.

Cenário	*Dataset*	*Total Number of Words*	*Number of Unique Words*	*Number of Comma*	*Number of Period*	*Number of Question Mark*
1	Train	935567	31034	86911	40356	124
	Dev	117229	10927	10889	4895	13
	Ref	58688	8245	5399	2367	4
	ASR	59106	8044	5576	2521	9
	IWSLT	320821	20309	15952	12590	1835
2	Train	935567	31030	86911	40480	-
	Dev	117229	10926	10889	4908	-
	Ref	58688	8245	5399	2371	-
	ASR	59106	8044	5576	2530	-
	IWSLT	320821	20306	15952	14425	-
3	Train	3752045	50000	351032	161119	-
	Dev	470841	22571	44416	20092	-
	Ref	235016	17400	21575	9837	-
	ASR	235136	16939	21934	9961	-
	IWSLT	320821	20306	15952	14425	-

Por fim, foram geradas as representações vetoriais das bases de dados. Para esta vetorização, foi utilizado o modelo que emprega o método GloVe de 300 dimensões, disponibilizado no repositório NILC-*Embeddings* [13], o qual faz parte do acervo do Núcleo Interinstitucional de Linguística Computacional (NILC) da Universidade de São Paulo (USP). Em seguida, os *embeddings*, por padrão, são alimentados em uma rede *highway* de duas camadas para aperfeiçoamento dos vetores. É utilizado também um mecanismo que decide realizar transformações nas representações geradas pelo GloVe ou mantê-las, criando, assim, *embeddings* mais sofisticados.

3.3 *Layers* densamente conectadas Bi-LSTM

Após o tratamento das bases de dados, a fase de construção dos modelos para treinamento é iniciada com a inserção das representações vetoriais em 4 (quatro) camadas densamente conectadas do tipo Bi-LSTM (*Bidirectional Long Short-Term Memory*), uma variação das Redes Neurais Recorrentes (RNN) tradicionais. Cada camada é composta por 50, 50, 50 e 300 neurônios, respectivamente, e utiliza a tangente hiperbólica (tanh) como função de ativação. Essa arquitetura processa dados sequenciais em ambas as direções, capturando informações contextuais tanto do passado quanto do futuro, permitindo que o modelo ofereça pontuações mais precisas.

3.4 *Self-attention* e *Conditional random field* (CRF)

A próxima fase do modelo é a realização de seu treinamento. Para isso, são implementadas duas abordagens a fim de maximizar a eficácia do modelo. A primeira é o mecanismo de atenção, comumente conhecido como *self-attention* e o segundo é uma camada CRF BRNN. Nas subseções a seguir, serão descritas em detalhes cada abordagem empregada.

Self-attention

Mecanismos de atenção, frequentemente utilizados em tarefas de tradução de textos, são técnicas que permitem ao modelo focar nas palavras mais relevantes ao fazer previsões. No contexto de restauração de pontuação, algumas palavras podem sugerir qual pontuação deve ser aplicada. Por exemplo, na frase "será que vai chover hoje", a palavra "será" é mais relevante que as demais, pois indica uma indagação, sugerindo que a frase deverá terminar com um ponto de interrogação. Este mecanismo de atenção funciona criando um vetor de contexto que corresponde à soma ponderada dos estados ocultos, conforme a Equação 2. Os pesos de atenção $\alpha_{t,i}$ são calculados a partir da Equação 1, onde $Score(Q_t, K_i)$ representa a similaridade entre a *query* Q_t e a *key* K_i, sendo normalizados pela função *softmax* para gerar os pesos de atenção.

A fórmula para o cálculo dos pesos de atenção $\alpha_{t,i}$ é dada por:

$$\alpha_{t,i} = softmax(Score(Q_t, K_i)) \tag{1}$$

. Onde:

- Q_t é o vetor de *query* (consulta), que representa a palavra no tempo t.
- K_i é o vetor de *key* (chave), que representa a palavra de índice i na frase.

A fórmula para o vetor de contexto c_t é dada por:

$$c_t = \sum_{i=1}^{T} \alpha_{t,i} h_i \tag{2}$$

. Onde:

- c_t é o vetor de contexto no tempo t.
- $\alpha_{t,i}$ é o peso de atenção entre o tempo t e a palavra i na sequência.
- h_i é o *hidden state* (estado oculto) ou vetor de valor associado à palavra i.
- T é o número total de palavras ou tokens na sequência de entrada.

Conditional Random Field (CRF)

Conditional Random Field (CRF) é uma técnica utilizada para modelar a probabilidade de uma sequência de rótulos Y ser gerada

a partir de uma sequência de palavras X, conforme apresentado na Equação 3, descrita por [14]. No contexto deste trabalho, uma camada CRF foi aplicada na etapa de construção do modelo de treinamento para a otimização da função de perda, utilizando a negação do resultado proveniente da função `crf_log_likelihood` do TensorFlow. O *log likelihood*, ou verossimilhança logarítmica, é a aplicação do logarítmo natural à função de verossimilhança, a qual, por sua vez, é uma métrica utilizada para medir quão provável é que um conjunto de parâmetros de um modelo probabilístico tenha gerado os dados observados.

A fórmula da probabilidade condicional no CRF é dada por:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, \mathbf{x_t})\right) \tag{3}$$

. Onde:

- $P(\mathbf{y}|\mathbf{x})$ é a probabilidade condicional da sequência de rótulos $\mathbf{y}$, dado a sequência de entrada $\mathbf{x}$.
- $Z(\mathbf{x})$ é o termo de normalização ou função de partição, que garante que a soma das probabilidades de todas as sequências possíveis seja igual a 1.
- K representa a quantidade de funções de características f_k utilizadas pelo modelo para capturar padrões relevantes nos dados.
- λ_k são os pesos ou parâmetros aprendidos durante o treinamento, associados a cada função de característica f_k.
- $f_k(y_t, y_{t-1}, \mathbf{x_t})$ são as funções de característica que dependem do rótulo atual y_t, do rótulo anterior y_{t-1}, e da entrada no tempo t, $\mathbf{x_t}$.

3.5 Avaliação de desempenho

A última etapa da metodologia proposta consiste na avaliação dos modelos treinados, para a qual são utilizadas as métricas ERR, SER, precisão, *recall*, *f1-score*, *train loss* e *perplexity* para compreender amplamente o resultado obtido a partir da fase de treinamento. A cada época treinada são realizadas predições utilizando os recortes das bases de dados *Ref* e *ASR*, realizando, assim, uma validação cruzada para que o modelo não se torne enviesado. A partir dessas predições, o modelo é avaliado utilizando as métricas mencionadas. As métricas ERR (Taxa de Erro de Restauração) e SER (Taxa de Erro de *Slot*) oferecem números que indicam a porcentagem de erro na restauração de pontuação, conforme as Equações 4 e 5.

$$ERR = 1 - \frac{\text{Número de } \textit{tokens} \text{ com pontuação restaurada corretamente}}{\text{Número total de } \textit{tokens} \text{ com pontuação restaurada}} \tag{4}$$

$$SER = \frac{\text{Número de } \textit{slots} \text{ incorretos (erros de inserção, omissão ou substituição)}}{\text{Número total de } \textit{slots} \text{ esperados no texto}} \tag{5}$$

O modelo também é avaliado utilizando as métricas precisão, *recall* e *f1-score*. Tais métricas são amplamente utilizadas para avaliar modelos de diferentes aplicações, pois oferecem uma visão abrangente de cenários. A precisão é ótima para cenários onde o custo de um falso positivo é alto; o *recall* é perfeito para aplicações onde é essencial que o modelo identifique todos os positivos; e o *f1-score* oferece um equilíbrio entre a precisão e o *recall*. Diferente das métricas ERR e SR, que se baseiam no número total de *tokens* com pontuação restaurada, essas métricas oferecem uma avaliação mais sofisticada, considerando três aspectos de uma predição (verdadeiros positivos, falsos positivos e falsos negativos), conforme descrito nas Equações 6, 7 e 8.

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos + Falsos Positivos}} \tag{6}$$

$$\textit{Recall} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos + Falsos Negativos}} \tag{7}$$

$$\textit{F1-Score} = 2 \times \frac{\text{Precisão} \times \textit{Recall}}{\text{Precisão} + \textit{Recall}} \tag{8}$$

Por fim, é importante mencionar as métricas *train loss* e *perplexity*. A primeira indica a diferença entre o *token* esperado e o *token* gerado pelo modelo, enquanto a métrica *perplexity* informa, basicamente, quão preparado o modelo estava ao receber uma determinada palavra. Essas métricas são essenciais para garantir que o modelo seja não apenas preciso em suas predições, mas também eficiente no processamento da base de dados, fornecendo uma visão mais ampla sobre o processo de aprendizado e adaptação do modelo aos dados.

4 RESULTADOS E DISCUSSÃO

Nesta seção, apresentam-se as informações necessárias para uma reprodução bem-sucedida dos experimentos, assim como a exposição dos resultados obtidos.

4.1 *Setup*

Para a implementação da metodologia apresentada, foram utilizadas diversas bibliotecas que viabilizaram o experimento. Na Tabela 3 a seguir, apresentam-se as bibliotecas utilizadas:

Para realizar o treinamento e inferência dos modelos, utilizou-se a plataforma Google Colab em sua versão mais recente. Ressalta-se que, para o treinamento do cenário 3, o qual utilizou 20% da base de dados, foi necessário o uso do plano Colab Pro, para garantir a utilização prolongada da GPU, visto que o limite do plano gratuito era atingido antes da finalização do treinamento, devido à grande quantidade de dados. A seguir, apresentam-se as informações técnicas da máquina utilizada para os experimentos:

- **IDE:** Google Colab
- **Placa Gráfica:** GPU Tesla T4
- **Processador:** Intel(R) Xeon(R) CPU @ 2.20GHz
- **Memória (HD):** 112.6 GB
- **RAM:** 15 GB
- **Versão do Cuda:** 12.2
- **Versão do driver cuDNN:** 535.104.05
- **Versão do interpretador Python:** 3.10.12

4.2 Treinamento do cenário 1

Na Figura 2, apresenta-se um gráfico com os resultados de *loss* e *perplexity* ao longo das 5 épocas de treinamento do modelo do cenário 1. É possível observar que a perda diminui a cada época, e a *perplexity* se estabiliza a partir da segunda época, indicando que o modelo, além de aprender a reduzir o erro à medida que é treinado, também está reconhecendo melhor os padrões nos dados utilizados.

Tabela 3: Bibliotecas utilizadas no estudo.

Nome	Versão	Descrição
datasets	2.20.0	Biblioteca para utilização facilitada de bases de dados.
numpy	1.26.4	Biblioteca para manuseio de *arrays* multidimensionais.
pandas	2.1.4	Biblioteca para manipulação e análise de dados.
sklearn	1.5.2	Biblioteca Python focada em algoritmos de Aprendizado de Máquina, funções para manipulação de bases de dados e outros.
tensorflow	2.17.0	Biblioteca de código aberto utilizada para atividades de aprendizado de máquina, focado principalmente em treinamento e inferência de redes neurais.
tqdm	4.66.5	Biblioteca para utilização de barra de progresso.
ujson	5.10.0	Biblioteca para codificação e decodificação de JSON.

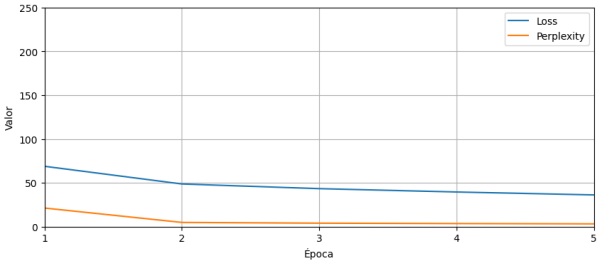

Figura 2: Comparação de *loss* e *perplexity* no cenário 1.

4.3 Treinamento do cenário 2

No segundo cenário proposto (ver Figura 3), o qual utiliza 5% da base de dados e não utiliza a classe <questionmark>, observam-se valores semelhantes de *loss* e *perplexity* do primeiro cenário. Porém, neste cenário, o valor de *perplexity* inicia quase oito vezes maior do que no cenário anterior, indicando que o modelo encontrou bastante dificuldade de identificar os padrões de sequências.

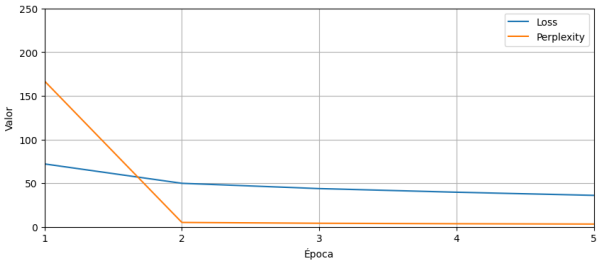

Figura 3: Comparação de *loss* e *perplexity* no cenário 2.

4.4 Treinamento do cenário 3

No último cenário proposto, conforme apresentado na Figura 4, os valores de *loss* apresentaram uma diminuição mais acentuada em comparação com os cenários anteriores, indicando que o modelo está realizando boas predições e, consequentemente, aprendendo melhor. Em relação à *perplexity*, observa-se que ela começa com um valor bastante elevado, mas diminui rapidamente, finalizando a quinta época com o valor de 2.37.

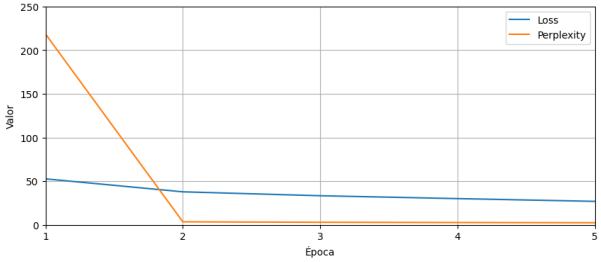

Figura 4: Comparação de *loss* e *perplexity* no cenário 3.

4.5 Análise dos Resultados

Nesta seção, são apresentados os resultados obtidos ao final da etapa de inferência dos modelos, sendo analisados os resultados *in-domain* e *cross-domain*.

Testes dos cenários *in-domain*

A fase de avaliação do modelo é crucial para entender sua adaptação à coletânea de dados e identificar quais ajustes podem ser implementados para elaborar um modelo mais completo e capaz de desempenhar uma determinada tarefa. Os resultados dos modelos nos três cenários propostos são apresentados nas Tabelas 4, 5 e 6, que correspondem, respectivamente, aos Cenários 1, 2 e 3. Nessas avaliações, foi utilizada a base de dados *Portuguese Legal Sentences v3*, ou seja, com o modelo sendo treinado e avaliado utilizando dados da mesma base, caracterizando, assim, uma avaliação *in-domain*. Ao analisar os resultados apresentados nas tabelas mencionadas, observa-se, em geral, uma melhoria das métricas, sendo o segundo cenário o de menor desempenho.

Ao comparar o primeiro e o segundo cenário, nota-se que, no segundo cenário, ao mapear todos os pontos de interrogação para pontos finais, a métrica de precisão diminuiu em até 3%, indicando

que o modelo foi impactado pelo contexto das frases terminadas por ponto de interrogação. No entanto, as métricas de *recall* e *f1-score*, tanto para vírgulas quanto para a avaliação geral, aumentaram em até 5%, mostrando que o modelo do segundo cenário compreendeu mais corretamente os contextos das frases e aplicou as pontuações de forma mais adequada. Além disso, no primeiro cenário, o modelo apresentou um desempenho de 0% nas métricas avaliadas para a classe <questionmark>, o que pode ser atribuído à sua baixa representatividade na base de dados.

Adicionalmente, ao comparar os resultados do segundo com o terceiro cenário, que contém uma quantidade maior de texto e, consequentemente, mais pontos e vírgulas, conforme apresentado na Tabela 2, observa-se um aumento em todas as métricas. É importante destacar que tal resultado era esperado, uma vez que, ao introduzir um conjunto de dados maior, o modelo adquire a habilidade de identificar mais padrões de frases, melhorando, assim, suas previsões.

Tabela 4: Métricas de desempenho do cenário 1 (*in-domain*).

	REF			ASR		
Pontuação	**Precisão**	***Recall***	***F1-score***	**Precisão**	***Recall***	***F1-score***
<comma>	68.08	43.71	53.24	70.07	43.95	54.02
<period>	65.14	42.65	51.55	61.87	42.08	50.09
<questionmark>	0.00	0.00	0.00	0.00	0.00	0.00
Média Geral	67.21	43.38	52.73	67.54	43.36	52.81
ERR		9.05%			9.27%	
SER		71.0%			70.4%	

Tabela 5: Métricas de desempenho do cenário 2 (*in-domain*).

	REF			ASR		
Pontuação	**Precisão**	***Recall***	***F1-score***	**Precisão**	***Recall***	***F1-score***
<comma>	64.43	49.95	56.28	65.55	50.75	57.21
<period>	62.25	42.31	50.38	60.75	42.47	49.99
<questionmark>	0.00	0.00	0.00	0.00	0.00	0.00
Média Geral	63.85	47.73	54.63	64.15	48.33	55.13
ERR		9.16%			9.36%	
SER		71.9%			71.0%	

Tabela 6: Métricas de desempenho do cenário 3 (*in-domain*).

	REF			ASR		
Pontuação	**Precisão**	***Recall***	***F1-score***	**Precisão**	***Recall***	***F1-score***
<comma>	69.64	55.95	62.05	70.63	56.12	62.55
<period>	65.96	49.33	56.44	67.29	49.27	56.89
<questionmark>	0.00	0.00	0.00	0.00	0.00	0.00
Média Geral	68.61	53.99	60.43	69.70	54.10	60.92
ERR		8.25%			8.24%	
SER		64.1%			63.1%	

Testes dos cenários *cross-domain*

Por outro lado, ao analisar os dados referentes à avaliação *cross-domain*, em que os modelos foram treinados com uma base de textos formais e avaliados utilizando o *dataset* IWLST, composto por transcrições de linguagem informal, observam-se resultados significativamente inferiores em comparação com os cenários *in-domain*.

Ao considerar o primeiro cenário, por exemplo, conforme a Tabela 7, a avaliação *cross-domain* alcançou uma média geral de desempenho de cerca de 35% menor em comparação ao mesmo modelo sendo avaliado em uma base formal (ver Tabela 4). Esse comportamento era esperado, visto que a discrepância entre o estilo dos textos de treinamento e de teste impacta diretamente a capacidade do modelo de generalizar para dados fora do domínio original.

Outra questão identificada refere-se ao desempenho crescente nas avaliações *cross-domain* (ver as Tabelas 7, 8 e 9), especialmente ao analisar a média geral de *recall* e *f1-score*, comportamento semelhante ao verificado nas avaliações *in-domain*. Isso comprova que o modelo utilizado no cenário 3, que mapeia os pontos de interrogação como pontos finais e adiciona uma maior quantidade de dados na etapa de treinamento, apresenta características mais adequadas à tarefa de restauração de pontuação em textos na língua portuguesa.

Além disso, ao analisar a métrica ERR nos três cenários propostos, observa-se que os valores são inferiores a 15%, o que indica que o modelo restaurou corretamente a maior parte das pontuações. No entanto, a métrica SER ultrapassou 100% em todos os cenários, evidenciando que os modelos inseriram um número de *tokens* incorretos superior ao total de *slots* de pontuação originalmente esperados. Isso significa que, além de restaurar pontuações corretamente, os modelos também adicionaram pontuações desnecessárias aos textos, resultando em um excesso de marcações.

Tabela 7: Métricas de desempenho do cenário 1 (*cross-domain*).

PONTUAÇÃO	**Precisão**	***Recall***	***F1-score***
<comma>	32.83	18.23	23.44
<period>	24.91	5.81	9.42
<questionmark>	0.00	0.00	0.00
Média Geral	30.86	11.98	17.26
ERR:		10.09%	
SER:		107.2%	

5 CONCLUSÃO

Neste trabalho, buscou-se realizar uma avaliação *in-domain* e *cross-domain* para a restauração de pontuação em dados textuais no idioma português. Para a realização desta pesquisa, foram investigados métodos disponíveis no estado da arte, além de selecionar bases de dados compostas por textos formais e informais. Com isso, foram conduzidos experimentos para avaliar a eficácia dos modelos de aprendizado implementados.

Tabela 8: Métricas de desempenho do cenário 2 (*cross-domain*).

PONTUAÇÃO	Precisão	*Recall*	*F1-score*
<comma>	27.43	24.24	25.74
<period>	18.98	9.10	12.30
<questionmark>	0.00	0.00	0.00
Média Geral	24.48	17.05	20.10
ERR:		11.74%	
SER:		124.8%	

Tabela 9: Métricas de desempenho do cenário 3 (*cross-domain*).

PONTUAÇÃO	Precisão	*Recall*	*F1-score*
<comma>	31.28	20.04	24.43
<period>	24.80	20.49	22.44
<questionmark>	0.00	0.00	0.00
Média Geral	27.80	20.25	23.43
ERR:		11.42%	
SER:		121.4%	

O método escolhido, originalmente projetado para a restauração de pontuação em textos escritos em inglês, foi adaptado para ser treinado e avaliado com a base de dados *Portuguese Legal Sentences v3*, composta por textos da área de direito. Adicionalmente, foi realizada uma avaliação *cross-domain* utilizando dados de 2014 a 2016 do *dataset* IWSLT. Os resultados mostraram que o modelo não foi capaz de identificar corretamente os pontos de interrogação, devido à baixa incidência dessa pontuação em comparação com outros sinais. No entanto, ao mapear os pontos de interrogação para pontos finais, o desempenho do modelo aumentou nas métricas de *recall* e *f1-score*. Em contrapartida, o modelo não conseguiu generalizar adequadamente para restaurar corretamente os sinais de pontuação em textos informais, sugerindo que uma abordagem de treinamento que combine textos formais e informais possa ser mais eficaz.

Portanto, a metodologia utilizada mostrou-se satisfatória no contexto desta pesquisa e os objetivos propostos foram atingidos. O modelo alcançou mais de 60% na métrica *f1-score* utilizando apenas 20% da base de dados e 5 (cinco) épocas na etapa de treinamento, ao lidar com textos formais em português. Isso indica que o caminho adotado é promissor para o desenvolvimento de um sistema de restauração de pontuação neste idioma, o que pode representar um avanço significativo na criação de textos mais corretos e de fácil compreensão em diversos contextos do cotidiano.

A fim de oferecer uma continuidade à pesquisa apresentada neste trabalho e alcançar resultados mais adequados em cenários formais e informais, propõem-se as seguintes melhorias no modelo de aprendizado: i) treinamento de modelos utilizando bases de dados que contenham textos formais e informais, a fim de aumentar a capacidade de generalização; ii) realização de testes com uma quantidade ainda mais expressiva de dados para verificar o ponto de saturação do modelo; e iii) treinamento de modelos em mais de 5 (cinco) épocas, a fim de verificar se o modelo pode continuar a melhorar com mais iterações.

AGRADECIMENTOS

O presente trabalho foi realizado com o apoio do Campus Manaus Zona Leste do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM).

REFERÊNCIAS

[1] Fatma Suliman, Manal Ben-Ahmeida, and Salma Mahalla. Importance of Punctuation Marks for Writing and Reading Comprehension Skills. *Faculty of Arts Journal*, 2019. doi: https://doi.org/10.36602/faj.2019.n13.06.

[2] Frederick Ungar. *Michaelis Dicionário Brasileiro da Língua Portuguesa.* Editora Melhoramentos Ltda, 2015.

[3] Maria Ramos, Rayssa Melo, Brenda de Souza, and Káritas de Deus. A pontuação textual como uma das maiores barreiras do desenvolvimento da escrita formal. *Revista EIXO*, 11(2):77–85, 2022. doi: https://doi.org/10.19123/eixo.v11i2.956.

[4] Ronghao Pan, José Antonio García-Díaz, and Rafael Valencia-García. Evaluation of transformer-based models for punctuation and capitalization restoration in spanish and portuguese. In Elisabeth Métais, Farid Meziane, Vijayan Sugumaran, Warren Manning, and Stephan Reiff-Marganiec, editors, *Natural Language Processing and Information Systems*, pages 243–256, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-35320-8.

[5] Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, et al. Paracrawl: Web-scale acquisition of parallel corpora. Association for Computational Linguistics (ACL), 2020.

[6] Tiago Barbosa de Lima, Luiz Rodrigues, Valmir Macario, Elyda Freitas, and Rafael Ferreira Mello. Automatic punctuation verification of school students' essay in portuguese. In *Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)*, pages 58–70. SBC, 2023.

[7] Tiago B De Lima, Pericles Miranda, Rafael Ferreira Mello, Moesio Wenceslau, Ig Ibert Bittencourt, Thiago Damasceno Cordeiro, and Jário José. Sequence labeling algorithms for punctuation restoration in brazilian portuguese texts. In *2022 11th Brazilian Conference(BRACIS)*, pages 616–630, 2022.

[8] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Mauro Cettolo, Christian Girardi, and Marcello Federico. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy, May 28–30 2012. European Association for Machine Translation. URL https://www.aclweb.org/anthology/2012.eamt-1.60.

[10] OBRAS. Projeto obras, corpo de obras brasileiras. Disponível em: https://www.linguateca.pt/OBRAS/OBRAS.html, 2013. Acesso em: 20 de novembro de 2024.

[11] Kai Luo. Punctuation restoration. Disponível em: https://github.com/k9luo/Punctuation-Restoration, 2020. Acesso em: 20 de novembro de 2024.

[12] Rufino de Melo. Portuguese legal sentences v3, 2023. URL https://huggingface.co/datasets/rufimelo/PortugueseLegalSentences-v3. Accessed: 21-08-2024.

[13] NILC. Repositório de word embeddings do nilc. Disponível em: http://nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc, 2017. Acesso em: 20 de novembro de 2024.

[14] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *UMass CRF Tutorial*, 2001. URL https://repository.upenn.edu/cis_papers/159/.