Extração Contextual com spaCy

Análise do Modelo: pt_core_news_sm

Matheus Mendonça dos Santos

André Santos de Oliveira

FACULDADE DE TECNOLOGIA DE JUNDIAÍ

Introdução

O que é PLN?

O Processamento de Linguagem Natural (PLN) é um campo interdisciplinar (Linguística, C. da Computação e IA) que tem como objetivo habilitar sistemas computacionais a compreender, interpretar e gerar a linguagem humana.

Extração Contextual (NER)

Uma tarefa de grande relevância no PLN. Focada em transformar dados textuais **não estruturados** (textos, artigos) em informação **estruturada** e semanticamente rica (ex: tabelas, bancos de dados).

Justificativa e Objetivo

Justificativa

A demanda por PLN em Português Brasileiro é alta (jurídico, médico, mídias sociais). Modelos genéricos falham em capturar as **nuances e particularidades** do idioma. Por isso, a validação de modelos específicos é uma necessidade premente.

Objetivo Geral

Avaliar na prática o desempenho e a adequação do modelo **pt_core_news_sm** da spaCy na tarefa de Reconhecimento de Entidades Nomeadas (NER) em um dataset de teste.

Revisão da Literatura

Evolução

As abordagens de NER migraram de métodos clássicos (baseados em regras) para modernas arquiteturas de **deep learning**, predominantemente os Transformers (Kaddour, et al., 2023).

BERTimbau

Um marco para o Português. (Souza, et al., 2020) demonstraram que modelos pré-treinados **exclusivamente** com dados brasileiros (BERTimbau) superam modelos multilíngues genéricos.

Desafios

Apesar dos avanços, desafios persistem: **ambiguidade semântica**, flexibilidade morfológica do português e a escassez de dados anotados de alta qualidade.

Procedimentos Experimentais: Materiais

Ferramentas e Modelo

- **Modelo:** pt_core_news_sm (v. 3.8.0)
- **Bibliotecas:** spaCy, pandas, scikit-learn, matplotlib
- **Ambiente:** Google Colab (Jupyter Notebook)

Dados e Avaliação

- **Gabarito (y_true):** Basefictícia.csv (35 linhas de anotações de entidades: PER, ORG, LOC, PROD).
- **Predições (y_pred):** O resultado do modelo spaCy.
- **Framework:** classification_report e confusion_matrix (scikit-learn).

Procedimentos Experimentais: Métodos

1. Carregar

Carregamento do Basefictícia.csv (gabarito **y_true**) com o pandas.

2. Processar

Execução do modelo pt_core_news_sm nas frases únicas para obter as predições (**y_pred**).

3. Alinhar

Criação de um novo DataFrame mesclando (merge) o gabarito (y_true) com as predições (y_pred).

4. Avaliar

Geração do classification_report e da confusion_matrix para análise dos resultados.

Resultados e Discussões

Resultados: Relatório de Classificação

A execução do modelo demonstrou um desempenho geral **fraco e inadequado**.

- **Acurácia (Accuracy):** Apenas 0.531 (Pouco mais da metade dos acertos).
- **Macro F1-score (Média):** Apenas 0.370.

Estes valores indicam um grave desequilíbrio e uma falha do modelo em manter precisão e recall, tornando-o inapropriado para a tarefa.

Resultados: Análise por Classe

PER (Pessoa)

Desempenho Perfeito
(F1-score: 1.0)
O modelo foi robustamente treinado para identificar nomes de pessoas.

LOC (Local)

Impreciso
(F1-score: 0.667)
Encontrou todos (Recall 1.0), mas classificou muitas coisas erradas como 'LOC' (Precision 0.5).

ORG (Organização)

Péssimo
(F1-score: 0.182)
Encontrou apenas 1 das 8 organizações (Recall 0.125).
Praticamente cego para esta classe.

PROD (Produto)

Falha Total
(F1-score: 0.0)
Totalmente incapaz. Não identificou **nenhuma** entidade desta categoria.

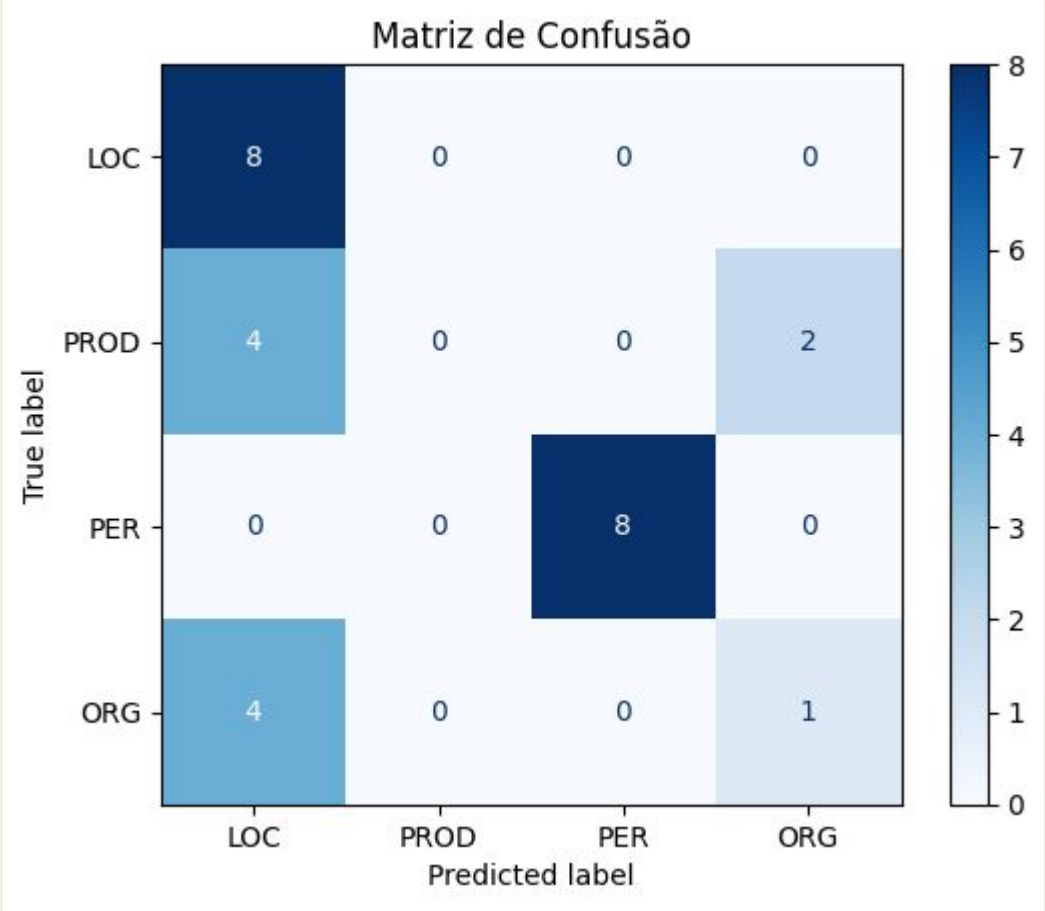
Resultados: Matriz de Confusão

Análise Visual dos Erros

A matriz confirma visualmente o relatório:

- **Eixo Y (True Label):** O que a entidade realmente era.
- **Eixo X (Predicted Label):** O que o modelo *achou* que era.

Principal Erro: O modelo confunde 'PROD' e 'ORG' (linhas 2 e 4), classificando-os erroneamente como 'LOC' ou 'PER'. Ele não tem confiança para prever 'PROD' ou 'ORG' e "chuta" as classes que conhece melhor.



Conclusão e Trabalhos Futuros

Conclusão

O modelo pt_core_news_sm é **inadequado** para a tarefa proposta.

Embora rápidos, modelos "pequenos" (sm) carecem da capacidade e do conhecimento contextual para lidar com entidades de domínio específico.
Para aplicações reais, é **imprescindível** o uso de modelos mais robustos (Transformers) e *fine-tuning*.

Trabalhos Futuros

- Testar modelos maiores (md, lg).
- Comparar com arquiteturas Transformers (ex: **BERTimbau**).
- Aplicar *Fine-Tuning* (ajuste fino) com dados de domínio específico.
- Validar em bases de dados reais, mais complexas e com maior volume.

Obrigado