Desenvolvimento de um sistema REN em textos jurídicos Brasileiros utilizando BERTimbau

Caio Túlio de Deus Andrade
Supervisor: Prof. Dr. Marcelo Finger

MAC 499 - Trabalho de Formatura Supervisionado
Instituto de Matemática e Estatística
Universidade de São Paulo

Janeiro 2022

Motivação

- Lawgorithm
- 1996 - “Estupro: Crime ou ‘Cortesia’?”- Prof. Dra Sylvia Pimentel
  - Vieses sexistas são externalizados nos textos dos autos
  - Análise Qualitativa
- 2019 - Pesquisa retomada com um viés quantitativo
  - Desenvolvimento de um sistema de REN capaz de identificar frases sexistas por meio de entidades nomeadas.
  - Elaboração de um conjunto de dados anotado por equipe especialista
  - Liberação dos dados não foi concluída a tempo

Introdução

- REN em textos jurídicos em Português
- BERTimbau: Versão em Português
- BERT
  - Verdadeiramente bidirecional
  - 2.5B palavras (Wikipedia) + 800M palavras (Book corpus)
  - Base do mecanismo de busca do Google
  - 110 Milhões de parâmetros (Base) / 340 Milhões de parâmetros
- Transferência de aprendizado
- Refinamento: especialização em uma tarefa

Tokenização - BERT

- Lista de embeddings interna: vocabulário
- Não é possível armazenar todas palavras internamente
- Solução: Algoritmo WordPiece.

Ex.: “Andando” -> “And”, “##an”, “##do”

Contexto

- O que é REN?
- Modelagem de linguagem impacta desempenho

Objetivo

- Estudar e documentar desafios em refinar modelos baseados em BERTimbau para REN

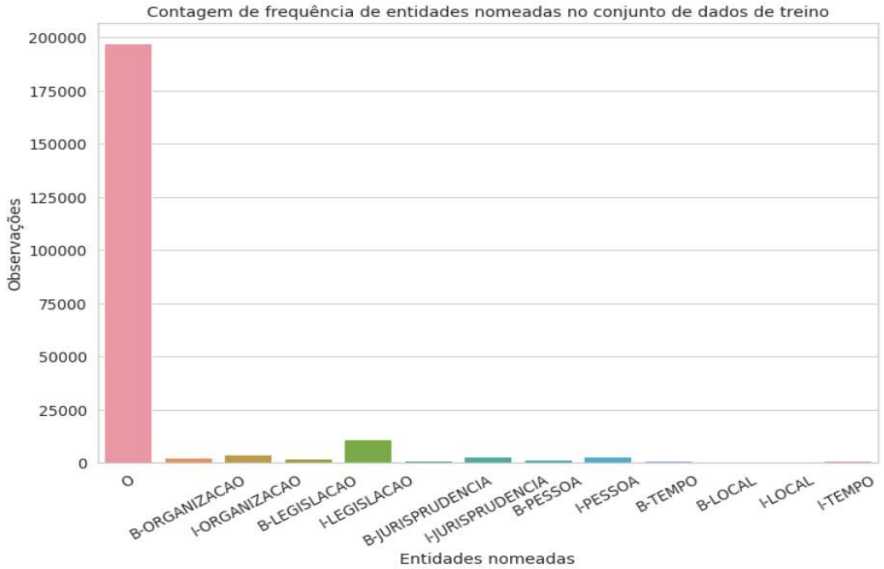
- Codificar um pipeline abstrato o bastante para ser reaproveitado

Conjunto de dados - LeNERBr

- Poucos conjuntos de dados de REN em texto jurídico
- 10395 Frases: 70% Treino, 15% Validação, 15% Teste.
- Anotado em IOB
- 6 entidades: Organização, Legislação, Jurisprudência, Pessoa, Tempo e Local
  - Totalizando 13 etiquetas

Conjunto de dados - LeNERBr

- Conjunto de dados desbalanceado
  - Comum em REN
  - F1 como métrica

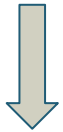


Readequação do conjunto de dados

Dados etiquetados palavra a palavra: BERT espera um input tokenizado a nível subpalavra

[“Entidade”, “Superior”, “Tribunal”, “Militar”]

O, B-ORG, I-ORG, I-ORG



“Entidade Superior Tribunal Militar”

[“Ent”, “##idade”, “Superior”, “Tribunal”, “Militar”]

O, IGNORE, B-ORG, I-ORG, I-ORG

Readequando etiquetas: consequência

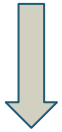
- Subtokens são ignorados ao calcular métricas de aprendizado (Loss e F1). Em consequência, não afetam aprendizado do modelo

- Mantidos na frase para uso do modelo de linguagem

Readequando etiquetas: consequência

[“Entidade”, “Superior”, “Tribunal”, “Militar”]

O, B-ORG, I-ORG, I-ORG



“Entidade Superior Tribunal Militar”

[“Ent”, “##idade”, “Superior”, “Tribunal”, “Militar”]

O, IGNORE, B-ORG, I-ORG, I-ORG

[“Entediado”, “o” “novo”, “juíz”]

O, O, O, O

“Entediado o novo juíz”

[“Ent”, “##edi”, “##ado”, “o”, “novo”, “juiz]

O, IGNORE, IGNORE O, ,O , O

Pipeline de treino

- Visando reaproveitamento de código, codificamos a pipeline em 4 classes, representando pontos cruciais de treino de modelo

1. Instanciação do modelo e tokenizador
2. Adaptação do conjunto de dados (retokenização e preprocessamento)
3. Treino
4. Avaliação

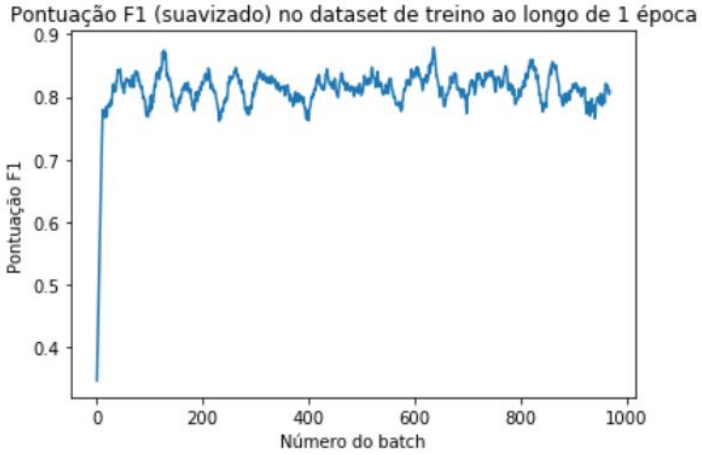
Experimento 1 - Refinamento

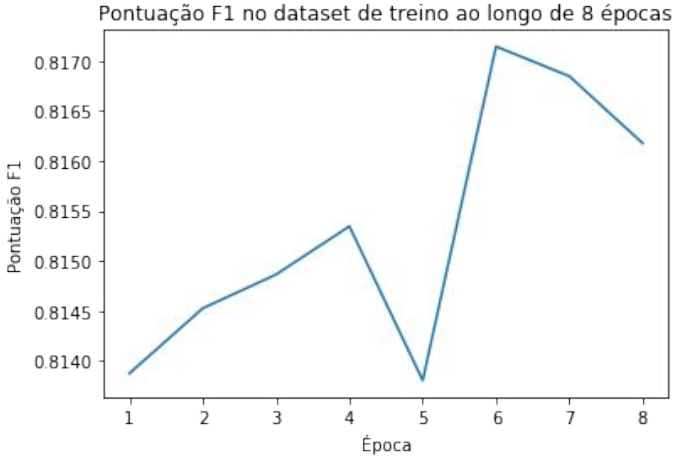
- Objetivo: validar abordagem e entender desempenho
  - Não houve tunagem de hiperparâmetros
- Parâmetros recomendados pelos autores do BERT:
  - Taxa de aprendizagem: 3e-4, 1e-4, 5e-3, 3e05
  - Otimizador: AdamW, Adam, SGD
  - Batch: 8, 16, 32, 64, 128

- Parametros escolhidos:
  - Taxa de aprendizagem: 3e-4
  - Batch: 8
  - Otimizador: AdamW

Experimento 1 - Refinamento

- Aprendizado rápido, saturação em poucos batches
- Comportamento similar no conjunto de teste
- Hipótese: poucas entidades nomeadas por batch (overfit em O)
- Melhoria futura: treino separado sem entidades O





Experimento 2 - Comparação de diferentes checkpoints

- Checkpoints comparados
  - Bertimbau base cased
  - Bertimbau large cased
  - Bert Base cased
  - Bert Base uncased
  - Bert Large cased
  - Bert Large uncased

- Esperavamos um desempenho superior de modelos em Português na variante Large -> maior poder computacional

Experimento 2 - Comparação de diferentes checkpoints

	checkpoint	f1_t	f1_e	loss_t	loss_e
1	neuralmind/bert-base-portuguese-cased	0.814	0.816	0.666	0.661
2	neuralmind/bert-large-portuguese-cased	0.812	0.816	0.691	0.66
3	bert-base-cased	0.811	0.814	0.681	0.668
4	bert-base-uncased	0.807	0.813	0.699	0.662
5	bert-large-uncased	0.807	0.813	0.708	0.664
6	bert-large-cased	0.811	0.812	0.692	0.66

Experimento 2 - Comparação de diferentes checkpoints

- Estrutura do texto jurídico dificulta desempenho de REN
- Inclusão de palavras específicas pode atenuar problemas para a tarefa

Conclusão

- Devido a retokenização, a tarefa de REN é particularmente difícil utilizando BERT.
- Pipeline desenvolvido oculta algumas dificuldades de adaptação.
- Natureza jurídica do texto pode impactar o desempenho.

Obrigado!