Mineração de texto voltada para artigos científicos

Rafaela C. dos S. Uchôas
Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo
São José dos Campos, Brasil

***Resumo*—Observando as dificuldades enfrentadas na pesquisa e catalogação de artigos científicos decorrentes da grande quantidade de documentos dispersos em diferentes bibliotecas digitais, percebe-se a necessidade de encontrar uma solução que agilize o processo de descoberta dos artigos mais relevantes para um determinado trabalho. As palavras-chave nem sempre são eficientes na busca, resultando em uma grande quantidade de documentos que precisam ser analisados manualmente. Nesse contexto, propomos o desenvolvimento de uma ferramenta que simplifica as informações básicas e acelera o processo de interpretação e pesquisa científica extraindo informações de arquivos de texto, focando em artigos científicos. A ferramenta preenche automaticamente os campos essenciais, como Título, Autores e identificador único. Os dados são extraídos de um banco de dados de artigos científicos e listados de forma organizada, facilitando a busca e classificação dos mesmos.**

***Index Terms*—Mineração de Texto, Extração de Texto**

I. Introdução

Observando algumas das dificuldades presentes na pesquisa e catalogação de artigos científicos, percebe-se que, devido à quantidade destes espalhados por diferentes bibliotecas digitais, a tarefa de fazer a gestão dos artigos para descobrir quais melhor se aplicam para um trabalho acaba se tornando difícil para o pesquisador.

Nessa, as palavras-chave nem sempre funcionam da maneira esperada na realização de certas pesquisas, já que a maioria dos resultados só identifica localização de um documento. Isso faz com que os pesquisadores percam muito mais tempo extraindo os dados necessários manualmente. Além disso, cada vez mais documentos são criados com o tempo, e em alguns casos a pesquisa de uma única palavra pode retornar uma quantidade absurda de resultados dos quais poucos são realmente desejados [1]. O processo de levantamento de artigos para uma revisão sistemática de literatura, por exemplo, é um bom exemplo de um trabalho que ainda é mais manual do que automatizado.

Diante desse problema, o Prof. Dr. Tiago Rodrigues Macedo sugeriu a ideia da criação de uma ferramenta de mineração de texto com o intuito de identificar esses dados de maneira mais fácil e organizada. Este trabalho consiste em criar uma ferramenta que, a partir de uma lista de artigos, retorna suas informações básicas e acelera o processo de interpretação dos dados de forma a facilitar o trabalho do pesquisador.

II. Conceitos Fundamentais

A. Artigo Científico

Um artigo científico é um documento escrito que descreve e comunica os resultados de uma pesquisa científica original. Geralmente, é publicado em revistas acadêmicas revisadas por pares ou apresentado em conferências científicas. Os artigos científicos seguem uma estrutura padrão e fornecem uma descrição clara e detalhada do problema de pesquisa, metodologia utilizada, resultados obtidos e conclusões tiradas. Aqui listamos alguns dos dados mais relevantes presentes num artigo científico:

- Título: O título do artigo fornece uma visão geral do conteúdo e é usado como referência para identificar o artigo.
- Autores: Os nomes dos autores do artigo são importantes para atribuir crédito e estabelecer a autoria do trabalho.
- Abstract: O resumo é um breve resumo do conteúdo do artigo e geralmente inclui os objetivos, metodologia, resultados e conclusões principais. Palavras-chave: Termos de indexação são usados para categorizar o artigo e facilitar a busca. Buscam abranger os principais conceitos e tópicos abordados no artigo.
- DOI ou identificador único: O DOI (Digital Object Identifier) é um identificador único atribuído a cada artigo científico. Ele permite que os artigos sejam referenciados de forma inequívoca e facilita a localização e o acesso ao conteúdo completo.
- Referências bibliográficas: As referências citadas no artigo fornecem informações sobre trabalhos anteriores relacionados, permitindo estabelecer a base teórica e contextual do trabalho atual.

B. Mineração de dados/texto

A mineração de dados/texto consiste no processo de extrair informações valiosas, padrões e conhecimentos úteis a partir de grandes conjuntos de dados. Combina técnicas de estatística, aprendizado de máquina, inteligência artificial e banco de dados com o objetivo de descobrir padrões, relações e tendências que não são facilmente identificados manualmente [2].

C. Processamento de linguagem natural (NLP)

O processamento de linguagem natural (NLP) é um componente da mineração de texto que realiza uma análise linguística que auxilia aos computadores no ato de entender, interpretar e manipular a linguagem humana. [23]

D. Reconhecimento de Entidade Nomeada (NER)

O Reconhecimento de Entidade Nomeada (NER) é uma técnica de processamento de linguagem natural que consiste

na identificação e categorização de entidades em textos. Essas entidades podem ser qualquer palavra ou série de palavras que se referem ao mesmo tema. Um modelo NER deve ser treinado para conseguir identificar e classificar entidades baseado no contexto. Uma biblioteca muito usada em NER em python é a spaCy, já que é uma biblioteca que oferece um sistema estatístico eficiente para esse campo, de forma a simplificar a aplicação de NLP. [15]

III. Trabalhos Relacionados

Para essa seção, foi definida uma estrutura em que os trabalhos lidos foram separados em três subdivisões correspondentes.

A. Métodos gerais de mineração de texto

De acordo com Aranha e Passos [4], a mineração de textos pode ser vista como uma extensão da mineração de dados ou da descoberta de conhecimento em bases de dados estruturadas. Aqui apresentam-se algumas das técnicas base que podem ser utilizadas:

1) Indexação: Faz uma rápida busca de documentos através de palavras-chave. Uma estrutura de dados de armazenamento inteligente proporciona aumento drástico de performance. Além de recuperar dados textuais, ela pode fazer cálculos com múltiplas palavras-chave de busca realizando uma ordenação segundo a avaliação de cada documento.
2) Processamento de Linguagem Natural (PLN): O processamento da linguagem natural (PLN) é outra técnica chave para a mineração de textos. Utilizando conhecimentos da área de linguística, o PLN permite aproveitar ao máximo o conteúdo do texto, extraindo entidades, seus relacionamentos, detectando sinônimos, corrigindo palavras escritas de forma errada e ainda desambiguizando-as. Participam normalmente na parte do pré-processamento dos dados, transformando-os em números.
3) Mineração de dados: As técnicas inteligentes de Mineração de dados (“Data Mining”) são muito úteis para atuar em cima de um banco de dados organizado e pré-processado. Dessa maneira, é possível identificar os conhecimentos relevantes da base de dados textual com técnicas como classificação e otimização.
4) O KDT [5], ou “Knowledge Discovery in Texts” refere-se ao processo de extração de informação útil (conhecimento) em documentos de textos não-estruturados. Esse processo utiliza abordagens já consagradas das áreas de Recuperação de Informação, Processamento de Linguagem Natural e Descoberta de Conhecimento em Banco de Dados como:
   - Técnicas de associação: A extração de regras de associação é uma técnica de Data mining que gera regras do tipo ”Se X Então Y”a partir de um banco de dados, onde X e Y são conjuntos de itens que ocorrem simultaneamente em várias instâncias.
   - Sumarização: O processo de sumarização seleciona as informações mais importantes do texto, tornando a descrição mais compacta, mas mantendo a mesma informação. É uma técnica bastante utilizada na mineração de textos com o intuito de identificar palavras ou frases mais importantes dos documentos.
   - Clusterização: As técnicas de clusterização são usadas para agrupar um conjunto de dados considerados similares em clusters ou grupos.

Além das técnicas citadas, uso de métodos não supervisionados para extração e organização de conhecimento recebe grande atenção por não exigirem conhecimento prévio dos dados. Alguns exemplos são o agrupamento particional por meio do k-means, também o agrupamento hierárquico onde se cria uma hierarquia de clusters que vão se mesclando sucessivamente com base na similaridade [6].

B. Extração de texto em PDFs

Um grande problema na extração de informações de artigos científicos aparece em razão do formato em que eles são disponibilizados. O PDF, ou ”Portable Document Format”, é um tipo de documento em que é muito difícil de extrair os dados devido ao seu formato focado em diagramação[7]. Esse tipo de formato prioriza as posições individuais dos caracteres e não leva em consideração informações semânticas importantes, como identificação de palavras, limite de parágrafos e, mais importante, os papéis semânticos, que identificam se o trecho pode ser classificado como título, fórmula, figura ou alguma outra estrutura.

O trabalho de Ramakrishnan, C., Patnia, A., Hovy, E. et al. [8] mostra a construção e desempenho de um sistema que extrai blocos de texto de artigos de pesquisa em formato PDF e os classifica em unidades lógicas com base em regras que caracterizam seções específicas, focando apenas no conteúdo textual dos artigos de pesquisa. Funciona em um processo de três etapas: 1- Detecção de blocos de texto usando processamento de layout espacial para identificar blocos de texto contíguos, 2- Classificação dos blocos de texto em categorias usando um método baseado em regras 3- Costura dos blocos de texto classificados em ordem correta, resultando na extração de texto de blocos agrupados por seção.

No trabalho de Basta e Basta [9], após os PDFs serem convertidos para arquivos de texto, a ideia comentada no parágrafo anterior é colocada em execução de forma adaptada. O primeiro subprocesso lê os artigos e transforma alguns termos; No subprocesso seguinte, é realizada a tokenização e limpeza (conversão para ‘lowercase’, por exemplo) do conteúdo dos arquivos; uso de N-gramas já que, no caso deste trabalho, estão sendo procuradas combinações de palavras; No último subprocesso, operadores de exemplos de filtragem são incluídos baseados na presença de atributos específicos.

C. Outras aplicações da extração de texto

No projeto de Álvarez e Alberto Cáceres [10] de uma ferramenta denominada FIP (Ferramenta Inteligente de Apoio à Pesquisa) para recuperação, organização e mineração de

grandes coleções de documentos, são utilizadas diversas técnicas de recuperação de informação, mineração de dados, visualização de informações e, em particular, técnicas de extração de informações, são usadas. Sistemas de extração de informação atuam sobre um conjunto de dados não estruturados e objetivam localizar informações específicas em um documento ou coleção de documentos, extraí-las e estruturá-las com o intuito de facilitar o uso dessas informações. O objetivo específico deste projeto é induzir, um conjunto de regras para a extração de informações de artigos científicos.

A pesquisa de Thakur e Kumar [11] mostra as principais técnicas utilizadas por vários autores em seus respectivos estudos de pesquisa relacionados à mineração de texto de literatura acadêmica. O LDA foi encontrado como a técnica mais utilizada pela maioria (29,5%) dos pesquisadores em seu trabalho de análise de texto, enquanto 17% dos pesquisadores utilizaram o algoritmo de agrupamento k-means, seguido da técnica de PNL por 3,4%, o método de Associação por 5,1% e a técnica SVM(Máquina de Vetores de Suporte) por 4,25% indicando que o LDA é amplamente empregado pelos pesquisadores, enquanto o algoritmo de Máquina de Vetores de Suporte é o menos frequente. Os resultados da pesquisa [12] revelam que a utilização da mineração de texto em diferentes campos acadêmicos aumentou significativamente ao longo do tempo. Foram identificadas áreas específicas de estudo em que a mineração de texto é aplicada ativamente. Além disso, foram encontradas palavras-chave que ocorrem em conjunto nos resumos de artigos sobre mineração de texto, permitindo identificar as principais palavras-chave em cada período de tempo com base em sua importância.

Com base na pesquisa [13], que revisa as aplicações de mineração de texto na pesquisa de inovação, foram definidas prioridades conceituais, metodológicas e contextuais para futuras pesquisas nessa área. Isso beneficia tanto iniciantes, que podem aproveitar algoritmos avançados para iniciar seus projetos, quanto usuários experientes, que podem lidar com as limitações e desafios das aplicações atuais para melhorar a qualidade da mineração de texto na pesquisa de inovação.

IV. Objetivos

A ideia principal é que a ferramenta seja capaz de pegar uma entrada de frases ou palavras chave e uma lista de artigos fornecida pelo usuário que, por exemplo, esteja desenvolvendo uma revisão sistemática de literatura. Baseada nessa lista o modelo irá identificar os campos mais importantes (Título, Autores e DOI) com os dados de cada artigo encontrado e lista-los mostrando a similaridade com a entrada inicial. Para isso, será usado o método de mineração de texto.

V. Metodologia Experimental

A implementação dessa ferramenta foi dividida em duas partes: A primeira, em que são identificadas informações importantes de cada artigo, é mais complicada e não tem muitos algoritmos que resolvem isso. Já a segunda, em que se calcula a similaridade entre o que o usuário inseriu e os artigos, já possui muitas implementações prontas na internet.

Com essas informações em mente, foi decidido treinar um modelo próprio que utiliza Reconhecimento de Entidade Nomeada (NER) para identificar o título, autores e DOI de cada artigo os identificando como entidades e definindo a posição de cada um nos textos de treino. Além disso o cálculo de distância escolhido para tratar a segunda parte da ferramenta foi a similaridade de cossenos, já que é um método que funciona melhor quando há uma disparidade grande no tamanho dos textos. Para tratar o texto pré-processado da segunda parte o modelo GloVe foi usado.

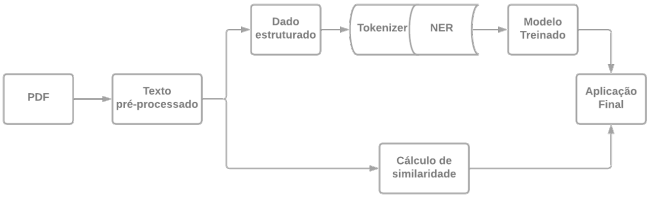


Figura 1. Diagrama de blocos do funconamento do pipeline experimental

A. Base de Dados

Para a construção da base de dados foram considerados artigos de acesso aberto em inglês, da plataforma ”PubMed”, e em português, do ”Portal de Periódicos CAPES”e do ”Scielo”.

Para a pesquisa em inglês, foram considerados artigos com as palavras-chaves ”Medicine; Neuroscience; Pattern Recognition”, ”Artificial Intelligence, Ethics, Prejudice”e ”Mathematics; Artificial Intelligence”com filtros focados em Ciência da Computação. Já para a base de artigos em português, foi necessário que a pesquisa fosse mais ampla: ”Inteligência Artificial”, ”Ética”e ”Medicina”com idioma em português.

Com isso, foi possível obter um arquivo ”.bib”com as informações referentes à cada artigo, o que possibilitou o download em massa dos PDFs através da plataforma ”Paperpile”.

Após a definição dos pdfs que compõem a base, foi feito o pré-processamento. Para isso, foram desenvolvidos scripts de conversão que leem todos os pdfs e convertem eles para txt atráves do uso da ferramenta ”PDFMiner”. Diferente do segundo script, ”conversor_similaridade.py”, que prepara os arquivos para o cálculo de similaridade, o primeiro script, ”conversor_entidade.py”, não trata ”stopwords”e nem lematiza os dados pois o objetivo é ter um texto puro que possa ser usado para identificação das entidades.

Um terceiro script, ”gera_training_data.py”, foi criado com o objetivo de fazer o cruzamento do texto dos PDFs com as informações do arquivo ”.bib”e separar as entidades no formato ideal para o treinamento (posição no texto e rótulo). Para diminuir problemas de falta de memória na hora do treinamento, os textos de cada artigo foram limitados para os 10 mil primeiros caracteres. Após esse passo, foi possível obter a seguinte relação de entidades por idioma:

Por último, o script ”to_spacy.py”converteu os arquivos dos dados para treinamento de ”.json”para ”.spacy”para que eles pudessem ser processados.

Artigos com entidades	Inglês	Português
Título	497	259
DOI	486	238
Autor	612	144
Total	**721**	**296**

Figura 2. Relação de artigos em que foram identificados cada tipo de entidade

B. Treinamento e Protocolo de Validação

Para o treinamento foi utilizada a versão 3.x da biblioteca spaCy. Nessa versão, todo o treinamento do modelo é feito através do prompt de comando e o gerenciamento dos parametros de treino são feitos através de um arquivo "config.cfg".

O protocolo de validação utilizado pelo spaCy é o "Hold-Out", que consiste na divisão dos dados de treino entre "dev"(para teste) e "train"(para treino). Para o treinamento desses modelos, os dados foram divididos em 80% para treino e 20% para teste.

O processo de treinamento é iterativo, de forma que a previsão do modelo é comparada com o rótulo para definir o gradiente de perda e atualizar os pesos do algoritmo. Com isso, é possível ajustar as previsões para que fiquem mais próximas do rótulo.

C. Principais dificuldades

É importante pontuar que houveram alguns obstáculos no desenvolvimento do modelo, principalmente na preparação da base de dados em português. Mesmo quando o idioma "Português"é selecionado na pesquisa, muitas vezes as bibliotecas online retornam artigos em espanhol ou inglês e a limpeza acaba tendo que ser manual. Além disso, muitas ferramentas de download ou conversão em massa não conseguem lidar com acentos e convertem o texto para Unicode, o que acaba obrigando o usuário a tratar o texto posteriormente. Por último, as ferramentas de classificação de entidade possuem bem menos opções de pipeline em português, o que pode afetar a acurácia do treinamento já que não é possível escolher que tipo de dado alimentou o pré-treinamento da pipeline.

O outro problema que influênciou bastante nos resultados foi a camada física encontrada no treinamento. No início, os modelos foram treinados localmente e a memória RAM não foi suficiente mesmo para treinos mais leves. O resultado variaram de memória excedida já no início do processo à horas de treinamento com acurácia zerada. Por isso, foi necessário fazer uso do ambiente "Google Colab"que possui maior RAM dedicada. Mesmo assim, ainda não foi possível utilizar as melhores configurações no treinamento do modelo, já que a memória era excedida.

VI. Resultados e Discussões

A. Modelo NER

O score é a pontuação total da pipeline que varia de 0 a 1. Ela é calculada baseada nos valores dos campos ENTS_F (F-Score), ENTS_P (Precisão) e ENTS_R (Recall). Os pesos de cada um desses campos são definidos no "config.cfg". Como padrão de treino do spaCy para NER, os valores bases para cada peso foram mantidos. Dessa forma, o único campo que tem influência direta na pontuação é o F-Score, que tem peso 1, enquanto os outros campos tem peso 0.

Inicialmente, os treinos do modelo foram definidos com toda a base de dados junta, com treino simultâneo dos três rótulos. Entretanto, a acurácia obtida dessa forma foi muito baixa, tendo o valor de 0.22 para o melhor modelo dos artigos em inglês e 0.17 dos artigos em português, como é possível ver nas figuras 3 e 4.

```
============================= Training pipeline =============================
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00    782.43    0.00    0.00    0.00    0.00
  0     200       5358.71  15198.30    0.00    0.00    0.00    0.00
  0     400       4299.09   1121.75   22.28   22.03   22.54    0.22
  0     600       1229.52    659.94   18.20   25.17   14.25    0.18
  0     800         60.25    526.86    1.51   26.09    0.78    0.02
  0    1000         52.83    581.95    3.71   41.67    1.94    0.04
  0    1200        120.82    537.02    9.36   16.89    6.48    0.09
  1    1400         43.47    515.90    9.64   27.78    5.83    0.10
  1    1600         64.96    518.62   15.93   28.81   11.01    0.16
  1    1800        194.67    528.61    8.44   29.46    4.92    0.08
  1    2000         89.72    533.07    7.96   32.71    4.53    0.08
```

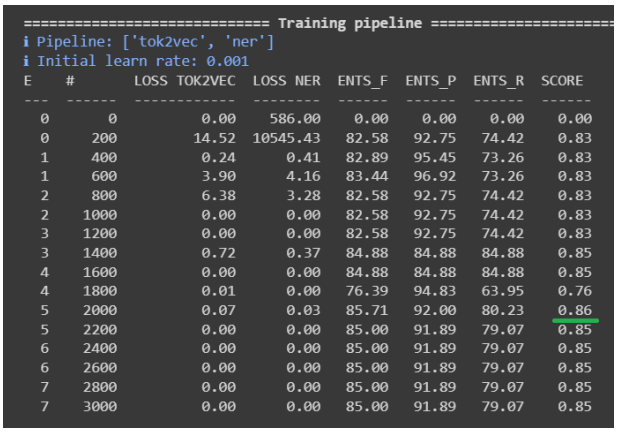
Figura 3. Treinamento dos rótulos em conjunto em inglês

```
============================= Training pipeline =============================
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00    684.79    0.00    0.00    0.00    0.00
  0     200       2319.36  13679.87    0.00    0.00    0.00    0.00
  0     400       5580.57   1199.12    0.00    0.00    0.00    0.00
  1     600       5678.16    860.94    0.00    0.00    0.00    0.00
  1     800       1284.00    550.25    3.08   27.27    1.63    0.03
  1    1000         60.77    565.99    0.00    0.00    0.00    0.00
  2    1200        135.07    533.02    4.90   25.00    2.72    0.05
  2    1400       1956.91    557.69    3.85   16.67    2.17    0.04
  3    1600      23071.04    692.29    9.06   12.62    7.07    0.09
  3    1800      28673.17    975.90    2.93   14.29    1.63    0.03
  3    2000       6797.69    582.86    4.74   18.52    2.72    0.05
  4    2200     179731.17   1110.61    3.06   25.00    1.63    0.03
  4    2400     136002.63    697.72   10.30   24.49    6.52    0.10
  5    2600      23277.70    722.02    2.84   11.11    1.63    0.03
  5    2800       6247.84    543.97    2.96   15.79    1.63    0.03
  5    3000    1500733.79   1332.06    1.06   25.00    0.54    0.01
  6    3200       3025.09    520.32    7.11   19.51    4.35    0.07
  6    3400       6506.88    587.42    5.48   17.14    3.26    0.05
  7    3600      16422.11    554.06    5.69   22.22    3.26    0.06
  7    3800       3006.40    489.19   16.89   22.32   13.59    0.17
  7    4000      38146.17    577.52    4.81   20.83    2.72    0.05
  8    4200        374.56    435.41    4.93   26.32    2.72    0.05
  8    4400      82092.89    592.68    4.00   25.00    2.17    0.04
```

Figura 4. Treinamento dos rótulos em conjunto em português

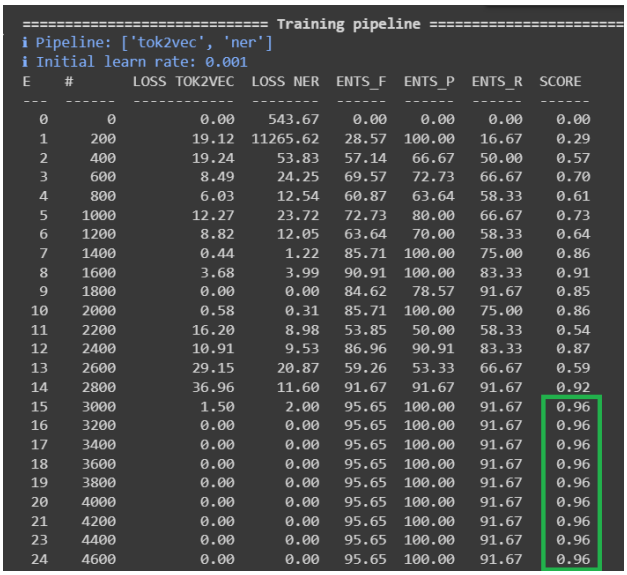
As figuras 5 a 10 mostram que ao realizar o treinamento com os mesmos parâmetros porém separando o treino de cada rótulo por vez o valor da pontuação aumentou significantemente.

Com esses dados, é possível comparar o valor dos melhores modelos para cada rótulo na figura 11.



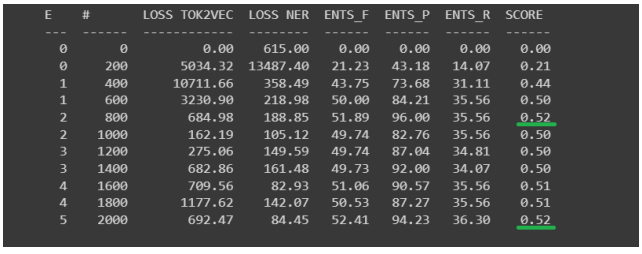
```
============================ Training pipeline ============================
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00    586.00    0.00    0.00    0.00    0.00
  0     200         14.52  10545.43   82.58   92.75   74.42    0.83
  1     400          0.24      0.41   82.89   95.45   73.26    0.83
  1     600          3.90      4.16   83.44   96.92   73.26    0.83
  2     800          6.38      3.28   82.58   92.75   74.42    0.83
  2    1000          0.00      0.00   82.58   92.75   74.42    0.83
  3    1200          0.00      0.00   82.58   92.75   74.42    0.83
  3    1400          0.72      0.37   84.88   84.88   84.88    0.85
  4    1600          0.00      0.00   84.88   84.88   84.88    0.85
  4    1800          0.01      0.00   76.39   94.83   63.95    0.76
  5    2000          0.07      0.03   85.71   92.00   80.23    0.86
  5    2200          0.00      0.00   85.00   91.89   79.07    0.85
  6    2400          0.00      0.00   85.00   91.89   79.07    0.85
  6    2600          0.00      0.00   85.00   91.89   79.07    0.85
  7    2800          0.00      0.00   85.00   91.89   79.07    0.85
  7    3000          0.00      0.00   85.00   91.89   79.07    0.85
```

Figura 5. Treinamento do rótulo DOI em inglês



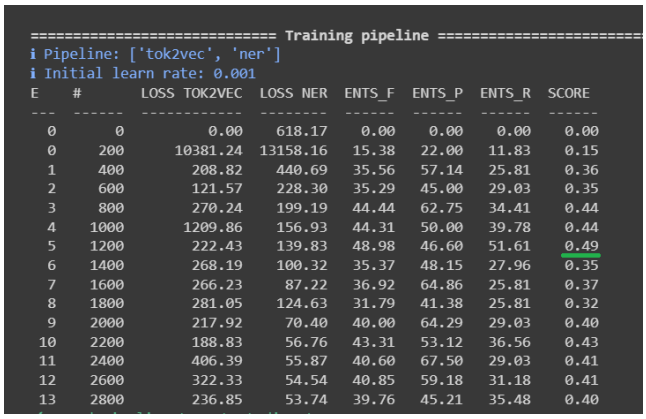
```
============================ Training pipeline ============================
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00    543.67    0.00    0.00    0.00    0.00
  1     200         19.12  11265.62   28.57  100.00   16.67    0.29
  2     400         19.24     53.83   57.14   66.67   50.00    0.57
  3     600          8.49     24.25   69.57   72.73   66.67    0.70
  4     800          6.03     12.54   60.87   63.64   58.33    0.61
  5    1000         12.27     23.72   72.73   80.00   66.67    0.73
  6    1200          8.82     12.05   63.64   70.00   58.33    0.64
  7    1400          0.44      1.22   85.71  100.00   75.00    0.86
  8    1600          3.68      3.99   90.91  100.00   83.33    0.91
  9    1800          0.00      0.00   84.62   78.57   91.67    0.85
 10    2000          0.58      0.31   85.71  100.00   75.00    0.86
 11    2200         16.20      8.98   53.85   50.00   58.33    0.54
 12    2400         10.91      9.53   86.96   90.91   83.33    0.87
 13    2600         29.15     20.87   59.26   53.33   66.67    0.59
 14    2800         36.96     11.60   91.67   91.67   91.67    0.92
 15    3000          1.50      2.00   95.65  100.00   91.67    0.96
 16    3200          0.00      0.00   95.65  100.00   91.67    0.96
 17    3400          0.00      0.00   95.65  100.00   91.67    0.96
 18    3600          0.00      0.00   95.65  100.00   91.67    0.96
 19    3800          0.00      0.00   95.65  100.00   91.67    0.96
 20    4000          0.00      0.00   95.65  100.00   91.67    0.96
 21    4200          0.00      0.00   95.65  100.00   91.67    0.96
 23    4400          0.00      0.00   95.65  100.00   91.67    0.96
 24    4600          0.00      0.00   95.65  100.00   91.67    0.96
```

Figura 6. Treinamento do rótulo DOI em português



```
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00    615.00    0.00    0.00    0.00    0.00
  0     200       5034.32  13487.40   21.23   43.18   14.07    0.21
  1     400      10711.66    358.49   43.75   73.68   31.11    0.44
  1     600       3230.90    218.98   50.00   84.21   35.56    0.50
  2     800        684.98    188.85   51.89   96.00   35.56    0.52
  2    1000        162.19    105.12   49.74   82.76   35.56    0.50
  3    1200        275.06    149.59   49.74   87.04   34.81    0.50
  3    1400        682.86    161.48   49.73   92.00   34.07    0.50
  4    1600        709.56     82.93   51.06   90.57   35.56    0.51
  4    1800       1177.62    142.07   50.53   87.27   35.56    0.51
  5    2000        692.47     84.45   52.41   94.23   36.30    0.52
```

Figura 7. Treinamento do rótulo título em inglês

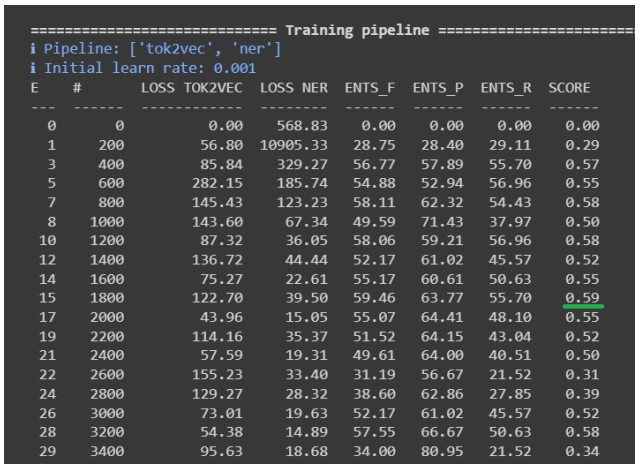


```
============================ Training pipeline ============================
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00    618.17    0.00    0.00    0.00    0.00
  0     200      10381.24  13158.16   15.38   22.00   11.83    0.15
  1     400        208.82    440.69   35.56   57.14   25.81    0.36
  2     600        121.57    228.30   35.29   45.00   29.03    0.35
  3     800        270.24    199.19   44.44   62.75   34.41    0.44
  4    1000       1209.86    156.93   44.31   50.00   39.78    0.44
  5    1200        222.43    139.83   48.98   46.60   51.61    0.49
  6    1400        268.19    100.32   35.37   48.15   27.96    0.35
  7    1600        266.23     87.22   36.92   64.86   25.81    0.37
  8    1800        281.05    124.63   31.79   41.38   25.81    0.32
  9    2000        217.92     70.40   40.00   64.29   29.03    0.40
 10    2200        188.83     56.76   43.31   53.12   36.56    0.43
 11    2400        406.39     55.87   40.60   67.50   29.03    0.41
 12    2600        322.33     54.54   40.85   59.18   31.18    0.41
 13    2800        236.85     53.74   39.76   45.21   35.48    0.40
```

Figura 8. Treinamento do rótulo título em português

```
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00    630.50    0.27    0.54    0.18    0.00
  0     200       1411.00  12456.35   53.32   50.82   56.08    0.53
  0     400         67.59    245.95   54.96   63.79   48.28    0.55
  1     600         72.01    183.41   61.01   66.88   56.08    0.61
  1     800        114.09    141.70   60.80   64.24   57.71    0.61
  2    1000        124.91    143.70   56.33   67.25   48.46    0.56
  2    1200         80.34     79.18   54.87   71.64   44.46    0.55
  2    1400        130.84     78.38   59.51   61.23   57.89    0.60
  3    1600         72.82     43.37   61.57   69.07   55.54    0.62
  3    1800        109.32     57.07   61.32   65.63   57.53    0.61
  4    2000        209.64     52.69   61.07   66.03   56.81    0.61
  4    2200        103.55     36.51   57.68   63.01   53.18    0.58
  4    2400        116.02     50.04   56.54   59.68   53.72    0.57
  5    2600         80.34     35.78   56.11   65.30   49.18    0.56
  5    2800         64.34     22.33   57.90   74.01   47.55    0.58
  6    3000        130.63     47.45   62.42   61.70   63.16    0.62
  6    3200         85.13     22.15   57.77   58.58   56.99    0.58
  6    3400        127.20     38.30   62.28   62.68   61.89    0.62
  7    3600        173.33     42.51   58.77   61.51   56.26    0.59
  7    3800        100.45     18.03   58.98   67.37   52.45    0.59
  8    4000         87.18     27.35   57.69   70.13   49.00    0.58
  8    4200         71.11     18.10   57.44   67.32   50.09    0.57
  8    4400         70.39     17.40   53.50   75.08   41.56    0.54
  9    4600        113.91     30.24   57.92   58.14   57.71    0.58
```

Figura 9. Treinamento do rótulo autor em inglês



```
============================ Training pipeline ============================
ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001
E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
---  ------  ------------  --------  ------  ------  ------  ------
  0       0          0.00    568.83    0.00    0.00    0.00    0.00
  1     200         56.80  10905.33   28.75   28.40   29.11    0.29
  3     400         85.84    329.27   56.77   57.89   55.70    0.57
  5     600        282.15    185.74   54.88   52.94   56.96    0.55
  7     800        145.43    123.23   58.11   62.32   54.43    0.58
  8    1000        143.60     67.34   49.59   71.43   37.97    0.50
 10    1200         87.32     36.05   58.06   59.21   56.96    0.58
 12    1400        136.72     44.44   52.17   61.02   45.57    0.52
 14    1600         75.27     22.61   55.17   60.61   50.63    0.55
 15    1800        122.70     39.50   59.46   63.77   55.70    0.59
 17    2000         43.96     15.05   55.07   64.41   48.10    0.55
 19    2200        114.16     35.37   51.52   64.15   43.04    0.52
 21    2400         57.59     19.31   49.61   64.00   40.51    0.50
 22    2600        155.23     33.40   31.19   56.67   21.52    0.31
 24    2800        129.27     28.32   38.60   62.86   27.85    0.39
 26    3000         73.01     19.63   52.17   61.02   45.57    0.52
 28    3200         54.38     14.89   57.55   66.67   50.63    0.58
 29    3400         95.63     18.68   34.00   80.95   21.52    0.34
```

Figura 10. Treinamento do rótulo autor em português

	Inglês	Português
Título	0.52	0.49
DOI	0.86	0.96
Autor	0.62	0.59

Figura 11. Comparação de pontuação dos modelos de cada rótulo

VII. Conclusões

Com os resultados obtidos do modelo, foi possível observar que a diferença de performance entre os modelos em português e em inglês não foram tão grandes. É provável que essa diferença seja pequena devido ao fato de que os dados de treinamento foram bem estruturados. É possível concluir, então, que o maior obstáculo na criação de modelos em português se dá no proceso de obtenção de dados para treino, o qual se mostra extenso, com escassez de 'databases' já prontas e, muitas vezes, exige bastante interferência manual.

Para os próximos passos dessa aplicação, seria interessante expandir a base de dados para treinamento do modelo em ambos os idiomas empregados. Com uma base mais ampla para treinamento será possível aumentar a acurácia do modelo e torná-lo uma ferramenta útil na organização de artigos para pesquisadores.

Referências

[1] "What is text mining, text analytics and Natural Language Processing?," What is Text Mining, Text Analytics and Natural Language Processing? Linguamatics, https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing (acessado 11 de Maio de 2023).

[2] google

[3] "Introdução — Machine learning — Google for developers," Google, https://developers.google.com/machine-learning/guides/text-classification?hl=pt-br (acessado 11 de Maio de 2023).

[4] C. Aranha and E. Passos, "A Tecnologia de Mineração de Textos," Revista Eletrônica de Sistemas de Informação, vol. 5, no. 2, 2006. doi:10.21529/resi.2006.0502001

[5] E. C. N. Barion and D. Lago, "Mineração de Textos," Revista de Ciências Exatas e Tecnologia, https://exatastecnologias.pgsscogna.com.br/rcext/article/view/2372 (acessado 2 de Maio de 2023).

[6] S. O. Rezende, R. M. Marcacini, and M. F. Moura, "O uso da mineração de textos para extração e Organização Não supervisionada de conhecimento.," Alice, http://www.alice.cnptia.embrapa.br/alice/handle/doc/895476 (acessado 11 de Maio de 2023).

[7] H. Bast and C. Korzen, "A benchmark and evaluation for text extraction from PDF," 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL), 2017. doi:10.1109/jcdl.2017.7991564

[8] C. Ramakrishnan, A. Patnia, E. Hovy, and G. A. Burns, "Layout-aware text extraction from full-text PDF of scientific articles," Source Code for Biology and Medicine, vol. 7, no. 1, 2012. doi:10.1186/1751-0473-7-7

[9] T. Basta, M. Basta, "Text mining classification of articles on social sustainability in supply chains," Indian Journal of Computer Science and Engineering, vol. 13, no. 5, pp. 1378–1387, 2022. doi:10.21817/indjcse/2022/v13i5/221305201

[10] A. C. Álvarez, Extração de Informação de Artigos Científicos: Uma Abordagem Baseada em Indução de Regras de etiquetagem, Mar. 2007. doi:10.11606/d.55.2007.tde-21062007-144352

[11] K. Thakur and V. Kumar, "Application of text mining techniques on scholarly research articles: Methods and Tools," New Review of Academic Librarianship, vol. 28, no. 3, pp. 279–302, 2021. doi:10.1080/13614533.2021.1918190

[12] H. Jung and B. G. Lee, "Research trends in text mining: Semantic Network and main path analysis of selected journals," Expert Systems with Applications, vol. 162, p. 113851, 2020. doi:10.1016/j.eswa.2020.113851

[13] D. Antons, E. Grünwald, P. Cichy, and T. O. Salge, "The application of text mining methods in innovation research: Current State, evolution patterns, and development priorities," R&D Management, vol. 50, no. 3, pp. 329–351, 2020. doi:10.1111/radm.12408

[14] PMC Open Access Subset [Internet]. Bethesda (MD): National Library of Medicine, https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/. (acessado 12 de Maio de 2023).

[15] "Uma Visão Geral sobre Named Entity Recognition (NER)", Medium.com, https://medium.com/elinttech/uma-visão-geral-sobre-named-entity-recognition-ner-4dc4e3b5e37a. (acessado 19 de maio de 2023).

[16] A. Nair, "Comparing documents with similarity metrics", Towards Data Science, 17-jan-2022, https://towardsdatascience.com/comparing-documents-with-similarity-metrics-e486bc678a7d. (acessado 19 de maio de 2023).

[17] A. Sieg, "Text Similarities: Estimate the degree of similarity between two texts", Medium, 04-jul-201, https://medium.com/@adriensieg/text-similarities-da019229c894. (acessado 28 de maio de 2023).

[18] G. Röhrich, "Find Text Similarities with your own Machine Learning Algorithm", Towards Data Science, 21-jun-2020, https://towardsdatascience.com/find-text-similarities-with-your-own-machine-learning-algorithm-7ceda78f9710. (acessado 28 de maio de 2023).

[19] J. Briggs, "BERT for measuring text similarity", Towards Data Science, 05-maio-2021, https://towardsdatascience.com/bert-for-measuring-text-similarity-eec91c6bf9e1. (acessado 28 de maio de 2023).

[20] S. Pal, "What is Text Similarity and How to Implement it?", MLSAKIIT, 10-nov-2021, https://medium.com/msackiit/what-is-text-similarity-and-how-to-implement-it-c74c8b641883. (acessado 28 de maio de 2023).

[21] Intuition Engineering, "Deep learning for specific information extraction from unstructured texts", Towards Data Science, 21-jul-2018, https://towardsdatascience.com/deep-learning-for-specific-information-extraction-from-unstructured-texts-12c5b9dceada. (acessado 18 de junho de 2023).

[22] U. Malik, "Python for NLP: Vocabulary and phrase matching with SpaCy", Stack Abuse, 21-mar-2019, https://stackabuse.com/python-for-nlp-vocabulary-and-phrase-matching-with-spacy/. (acessado 18 de junho de 2023).

[23] V. Reddy, "Build a Custom NER model using spaCy 3.0", Turbolab Technologies, 11-nov-2021, https://turbolab.in/build-a-custom-ner-model-using-spacy-3-0/. (acessado 18 de junho de 2023).

[24] "O que é processamento de linguagem natural?", Sas.com, 14-ago-2018, https://www.sas.com/pt_br/insights/analytics/processamento-de-linguagem-natural.html. (acessado 23 de junho de 2023).

[25] , https://www.kaggle.com/code/adepvenugopal/nlp-text-similarity-using-glove-embedding (acessado 26 de junho de 2023).