Relatório 24 - Prática: Processamento de Linguagem Natural (NLP) (III)

Lucas Augusto Nunes de Barros

Descrição das atividades

O card propõe a execução de duas atividades, um minicurso sobre processamento de linguagem natural oferecido pela equipe do TensorFlow na plataforma do Youtube, e ainda no mesmo ambiente, o outro vídeo proposto é do canal FreeCodeCamp, que aborda de forma mais abrangente os aspectos práticos do processamento de linguagem natural.

1. *Natural Language Processing (NLP Zero to Hero)*

O processo de *tokenização* no processamento de linguagem natural consiste na divisão de um texto em unidades menores, denominadas *tokens*, que representam palavras, pontuações e outros elementos linguísticos de forma numérica, permitindo que sistemas computacionais analisem e processem a informação textual de forma estruturada. A *tokenização* é realizada pelo componente *tokenizer*, que aplica regras linguísticas específicas para a língua especificada, segmentando o texto com base em espaços, pontuações e características morfológicas. Por exemplo, ao processar a frase "Eu amo meu doguinho.", o *tokenizer* gera os tokens:

"Eu",

"amo",

"meu",

"doguinho"

"."

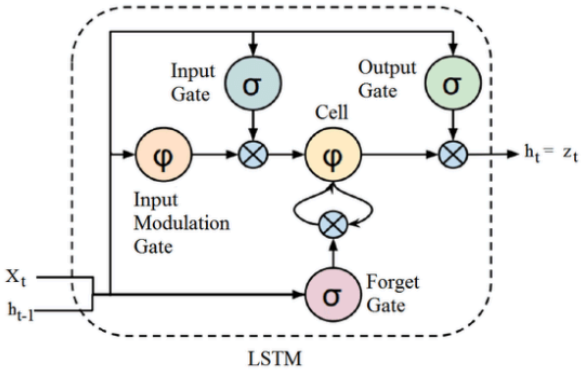
Esses *tokens* são armazenados no objeto e servem como base para análises futuras, facilitando a representação numérica do texto.

Os *tokens* OOV e a função *pad_sequences* são elementos essenciais na preparação de textos para modelos de aprendizado de máquina. Os *tokens* OOV representam palavras ausentes do vocabulário construído durante o treinamento de *tokenização*, essas palavras desconhecidas são então mapeadas para um índice especial que permite processar textos novos sem erros. A função *pad_sequences* uniformiza o comprimento das sequências de *tokens* geradas a partir dos textos, conforme exigido por redes neurais, que possuem entradas de tamanho fixo.

O aprendizado de máquina no processamento de linguagem natural é amplamente utilizado para modelar e interpretar textos. As redes neurais recorrentes possuem um papel central nesse tipo de tarefa, pois necessitam da compreensão de sequências. As redes neurais recorrentes são projetadas para capturar dependências temporais em dados sequenciais, permitindo que o modelo processe palavras em contexto e retenha informações de palavras anteriores. No processamento de

linguagem natural, essas redes são aplicadas em tarefas como tradução automática, análise de sentimentos e geração de texto, onde a ordem das ocorrências pode interferir no significado final da frase. A partir do treinamento redes neurais recorrentes aprendem padrões linguísticos, ajustando seus pesos para prever a relação entre palavras, o que as torna ferramentas poderosas para lidar com a complexidade e a variabilidade da linguística.

Uma das arquiteturas de redes neurais recorrente (RNN) mais utilizada é a Long Short-Term Memory (LSTM) um tipo de RNN projetada para resolver o problema das dependências de longo prazo em dados sequenciais. As LSTM são eficazes em tarefas que exigem a memorização de informações por longos períodos. A principal inovação dessa arquitetura é sua capacidade de manter e atualizar informações por períodos de tempo através de três portas principais: porta de entrada, porta de esquecimento e porta de saída. Essas portas regulam o fluxo de informações dentro da célula de memória, permitindo que a rede “lembre” e “esqueça” informações importantes.



LSTM

2. *Natural Language Processing with spacy & Python - Course for Beginners*

O spacy é uma biblioteca de código aberto em Python voltada para o Processamento de Linguagem Natural (PLN), projetada para oferecer alta performance e facilidade de uso em projetos reais. Suporta diversas tarefas, como *tokenização*, reconhecimento de entidades nomeadas e análises sintáticas, sendo amplamente utilizada em aplicações industriais e acadêmicas. Seus modelos pré-treinados, disponíveis para múltiplas línguas, incluindo o português, integram redes neurais otimizadas, permitindo processamento rápido e preciso de textos.

A biblioteca opera por meio de *pipelines* de processamento, no qual o texto é sequencialmente analisado por diferentes componentes que geram rótulos armazenados em containers. Esses containers organizam as informações de forma hierárquica, facilitando o acesso e a manipulação dos atributos linguísticos.

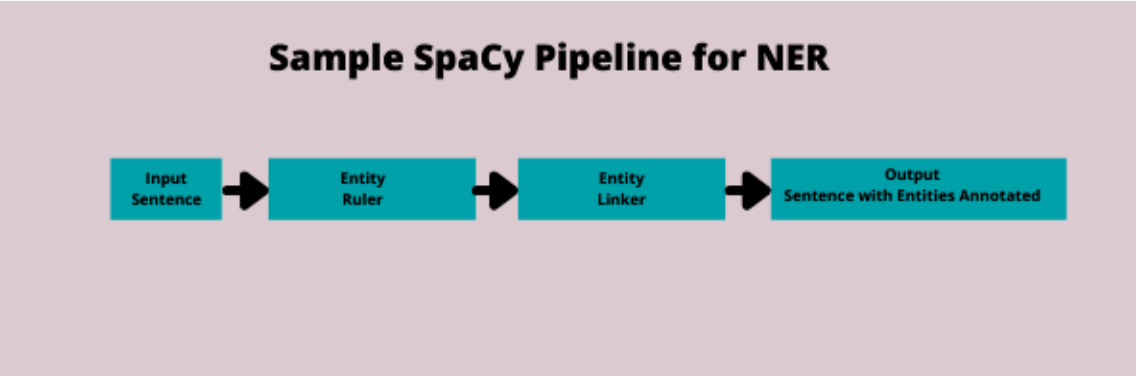
Neste contexto, os atributos são as propriedades dos objetos do pipeline que contêm informações linguísticas geradas durante o processamento do texto. Esses atributos são preenchidos automaticamente pelos componentes do pipeline e seguem regras definidas pelos modelos. As regras de atributos determinam como a biblioteca irá trabalhar com essas propriedades com base em modelos estatísticos, regras linguísticas e contexto textual.

As regras de atributos fazem referência ao conjunto de lógicas baseadas em *machine learning* e regras linguísticas utilizadas pela biblioteca para determinar os valores de cada atributo de cada objeto.

O mini curso ainda aborda técnicas mais avançadas para identificar e classificar elementos dentro de textos, utilizando ferramentas como expressões regulares para reconhecer padrões complexos. Essas expressões permitem a detecção de sequências de caracteres, oferecendo uma abordagem flexível para capturar estruturas que podem variar em comprimento ou composição. Essa técnica é particularmente valiosa em cenários onde os modelos não identificam corretamente certos padrões linguísticos.

A customização de vocabulários representa outra estratégia no processamento de textos, permitindo que sistemas adaptem-se à variabilidade linguística. A integração de componentes personalizados em pipelines de processamento de linguagem natural oferece uma abordagem mais robusta para a interpretação de textos, combinando conhecimentos pré definidos com regras ajustadas manualmente para as situações mais específicas e aplicações com vocabulários mais técnicos.

No contexto da biblioteca spacy, a sigla NER refere-se a *Named Entity Recognition*, Reconhecimento de Entidades Nomeadas em tradução livre, uma técnica aplicada no processamento de linguagem natural para identificar e classificar entidades específicas em um texto, como nomes próprios, locais e datas.



Por fim, o mini curso aborda uma aplicação do processamento de linguagem natural em texto do mercado financeiro, utilizando o modelo pré-treinado do spacy para separar nomes de companhias, bem como reconhecer suas siglas e significados através de dados em arquivos csv.

Conclusão

As atividades propostas destacam as capacidades de ferramentas no processamento de linguagem natural. A biblioteca spacy se mostra uma ferramenta robusta para tarefas como *tokenização*, reconhecimento de entidades nomeadas e análise de dependências, bem como oferece flexibilidade através de modelo pré treinados e a possibilidade de adaptar o pipeline para necessidades específicas. A integração de redes neurais recorrentes, especialmente as arquiteturas LSTM e a biblioteca spacy, proporciona uma base robusta para o desenvolvimento de modelos capazes de compreender tanto a estrutura quanto a semântica de textos.

Referências

[1] <https://spacy.io/usage>

[2] <https://python-textbook.pythonhumanities.com/03_spacy/03_01_03_linguistic_annotations.html>

[3] <https://python-textbook.pythonhumanities.com/03_spacy/03_01_04_pipelines.html>

[4] <https://medium.com/@sujathamudadla1213/module-5-lecture-5-2-4d650a766f0e>

[5] <https://www.datageeks.com.br/lstm/>

[6] <https://www.deeplearningbook.com.br/arquitetura-de-redes-neurais-long-short-term-memory/>

[7]

[8]