Luiz Felipe Nagatani Rabello

Relatório LAMIA - Módulo 22

Curso tensorflow

O curso se inicia com conceitos de NLP já abordados em cards anteriores, como tokenização, porém se diferencia ao mostrar como funciona sequencing, padding na sequences e por mostrar um classifier de texto, até agora NLP havia sido demonstrado somente teoricamente de forma rasa e com poucas práticas

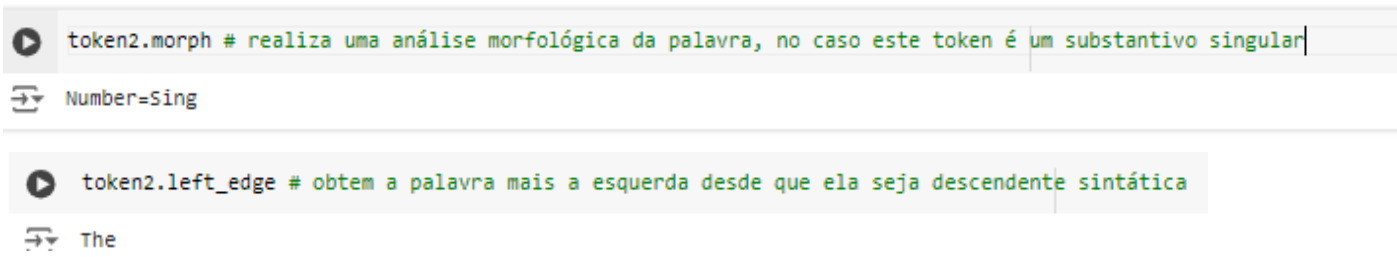
São apresentadas redes neurais recorrentes, onde cada nodo da rede possui uma conexão para a camada seguinte e para si mesmo, fazendo que dados de execuções anteriores do mesmo nodo possibilitando que informações obtidas anteriormente sejam utilizadas na nova execução, tornando possível processamento de linguagem natural com contexto.

Para casos onde a palavra que dá o contexto está muito longe da palavra a ser processada é necessario utilizar uma arquitetura chamada LSTM, que introduz uma estrutura nova chamada cell state, passando os dados da execução anterior não somente para a proxima camada, mas para a rede inteira com o passar das épocas, a informação pode ser passada tanto para a camada seguinte quanto para a anterior

Minicurso SpaCy

O vídeo se inicia com uma breve apresentação dos conceitos de NLP e containers, seguido de um curto guia de como instalar o SpaCy em seu computador, o SpaCy é uma biblioteca do python utilizada para facilitar o processamento de linguagem natural.

Em seguida são apresentados alguns comandos para processar textos, a tokenização é ainda mais facil do que a executada no tensorflow, sendo fácil de manipular e consultar frases dentro de um texto, uma função nova apresentada foi a possibilidade de analisar a morfologia de uma palavra e seus descendentes sintáticos.



É possivel analisar frases como vetores de palavras e calcular similaridade entre elas utilizando o .similarity, que retorna um valor entre 0 e 1, sendo 1 um vetor identico e 0 um vetor completamente diferente.

```
[17]
print(doc1, '<->', doc3, doc1.similarity(doc3))
```

```
I like salty fries and hamburguers. <-> The Empire State Building in is New York. 0.1840035527750666
```

```
doc4 = nlp('I enjoy oranges.')
doc5 = nlp('I enjoy apples.')

print(doc4, '<->', doc5, doc4.similarity(doc5))
# não retratam a mesma fruta, porém são semelhantes por possuirem a mesma estrutura
```

```
I enjoy oranges. <-> I enjoy apples. 0.9775702131220241
```

É possível que entre a criação do minicurso e o momento atual, o modelo en_core_web tenha sido atualizado, pois a seção de entity ruler apresenta técnicas para contornar casos onde o tipo do token não é o correto, no vídeo o nome de um local foi classificado como nome de uma pessoa, porém na execução da prática não foi necessário aplicar o entity ruler pois o modelo conseguiu compreender que o nome do local era uma GPE.

```
[1] import spacy
```

```
[2] nlp = spacy.load('en_core_web_sm')
    text = 'West Chestertenfieldville was referenced in Mr. Deeds.'
```

```
[3] doc = nlp(text)
```

```
for ent in doc.ents:
    print(ent.text, ent.label_) #assume que west Cherster... é um local e Deeds é uma pessoa por conta do contexto fornecido pelo título "Mr."
```

```
West Chestertenfieldville GPE
Deeds PERSON
```

Uma ferramenta interessante do SpaCy é o matcher, que é capaz de analisar o texto e buscar uma substring que satisfaça uma condição proposta, um exemplo apresentado no curso é a detecção de emails dentro de um texto.

```
for match in matches:
  print(match, doc[match[1]:match[2]])
```

```
(16571425990740197027, 6, 7) wmattingly@aol.com
(16571425990740197027, 13, 14) mathally@yhoo.com
```

Para Fins de experimentação foi utilizado como dataset um trecho na wiki de The binding of isaac falando sobre a quantidade de itens adicionados em cada atualização, foi necessário o LIKE_NUM como pattern para que fosse extraído do texto somente os números, após isso o array matcher foi percorrido através de iteração, somando o número de itens adicionados por atualização até chegar no número da atualização mais recente 719.