Processamento de Linguagem Natural em artigos científicos

Bruno Cesar Puli Dala Rosa, 95378
Cristofer Alexandre Oswald, 94509

Inteligência Artificial
Prof. Dr. Wagner Igarashi
Ciência da computação, UFM 2018

Introdução

O Processamento de Linguagem Natural (PLN) sempre foi um grande desafio para a computação. Programar o computador para entender linguagem natural não é uma tarefa trivial.

Os avanços na área de IA possibilitam o surgimento de tecnologias para PLN cada vez mais avançadas e úteis em diversos contextos de aplicação

Introdução

Neste trabalho buscamos desenvolver uma ferramenta de PLN com o foco de processar artigos científicos extraindo destes suas informações cruciais.

O avanço de técnicas como esta possuem como objetivo a sofisticação e a aceleração ao processo de análise automatizada de textos científicos.

O problema

Para o propósito deste trabalho, foi requisitado a extração de informações de textos científicos. Para isso, o problema foi dividido em 3 etapas:

Leitura do texto de forma crua;

Pré-processamento, que pode ser dividido como remoção de caracteres e palavras inválidas/inúteis e tokenização (separação da string em lista de palavras);

Extração das informações

Requisitos

A ferramenta desenvolvida recebe como entrada arquivos do tipo pdf e os processa retornando em formato de texto os seguintes dados:

- **Os 10 termos mais citados no texto do artigo**
- **As Referências bibliográficas**
- **A instituição e autores**

Estrutura do programa

O software desenvolvido foi implementado na linguagem Python utilizando as seguintes bibliotecas:

- ➔ **Textract**
- ➔ **Re**
- ➔ **Nltk**

A implementação, depuração e o teste desta ferramenta foram realizadas em um computador Intel I5 com 8 Gb de memória sob o sistema operacional Ubuntu 18

Textract

Realiza o processamento inicial do arquivo, extraindo o texto em sua forma crua, em apenas uma string contínua.

Re

Ferramenta para a geração de Expressões Regulares

Foi utilizada para gerar uma expressão regular que remove caracteres inválidos/inúteis de textos

Nltk

Responsável por dividir o texto em tokens, ou seja, uma lista de palavras, sendo cada palavra um token. Além disso, fornece uma lista de palavras que não agregam significado ao texto como “I”, “the”, “of” e pontuações como “.”, “<”, “!”.

Extração dos dados

Para cada um dos requisitos, foi modelo um algoritmo capaz de extrair os dados, sendo cada um deles feito de maneira específica para os artigos dados.

Termos mais citados

A extração do 10 termos mais citados é feita pela simples contagem das palavras, visto que o texto já está pré-processado. É importante notar que, ao encontrar o início das referências, a contagem para. Além disso, dígitos e palavras com duas letras ou menos foram ignorados.

```
top_words = {}

for word in keywords:
    word = word.lower()

    if (not word.isdigit()) and (len(word) > min_tam):
        if word in refs:
            break
        else:
            if word not in top_words.keys():
                top_words[word] = 1
            else:
                top_words[word] += 1

sorted_top = sorted(top_words.items(), key=lambda x: x[1])
sorted_top = sorted_top[-10:]
```

Neste algoritmo, é criado um dicionário, onde as chaves são as palavras e o valor é a contagem delas. Com esse dicionário, iteramos sobre as palavras e fazemos a contagem. Nota-se que a contagem é terminada quando se encontra o início das referências.

Após a contagem é feita a ordenação do dicionário e retirada as 10 últimas palavras e seus valores (visto que a ordenação é feita em ordem crescente).

Extração das referências

Para as referências, utilizamos o texto cru, visto que o pré-processamento dificulta a sua extração. Com a string que representa o texto, procuramos o título referente as referências e extraímos todo o texto a partir de então.

```
ref_pos = text.find('REFERENCES')
if ref_pos == -1:
    ref_pos = text.find('EFERENCES')

if ref_pos != -1:
    ref_text = text[ref_pos-1:]
```

Execução silenciosa

Para a execução do programa, é necessário instalar as bibliotecas citadas, as quais, para a plataforma utilizada, foram instaladas com o software *pip*. Com o sistema configurado, basta apenas utilizar uma chamada python para inciar a execução: “python read.py”

O programa implementado trabalha sob a base de artigos salvas no diretório de entrada do projeto. Com a conclusão do processamento de cada artigo um arquivo de texto (txt) é criado no diretório de saída do projeto e as informações extraídas são escritas nestes arquivos de saída. Assim, o programa funciona com a execução de apenas um comando, sem a necessidade de fornecer uma entrada e a saída, como já citado, é feita em forma de arquivo.

Referências

Textract - http://textract.readthedocs.io/en/stable/

PDFs em Python - https://medium.com/@rqaiserr/how-to-convert-pdfs-into-searchable-key-words-with-python-85aab86c544f