APRENDIZADO DE MÁQUINA PARA ANÁLISE SEMÂNTICA DE ARTIGOS CIENTÍFICOS

Aluno: Éverton Cardoso Acchetta,
Orientador: Prof. Dr. Paulo Sergio Silva Rodrigues[2]
[1,2] Ciência da Computação, Centro Universitário FEI
unieeacchetta@fei.edu.br e psergio@fei.edu.br

Resumo: A obtenção na internet de artigos científicos de um assunto específico ainda é uma tarefa trabalhosa para pesquisadores, uma vez que a quantidade de base de dados tem crescido de maneira volumosa, refletindo a importância e o interesse da produção científica, mas ao mesmo tempo dificultando a recuperação e organização de informações. Neste projeto, é apresentada uma proposta para indexação, recuperação, organização e seleção automática de informação de artigos acadêmicos-científicos utilizando como fonte o site *Semantic Scholar*.

1. Introdução

Tradicionalmente, quando deseja-se pesquisar em uma determinada área acadêmica, uma das primeiras tarefas que deve ser feita é o levantamento bibliográfico para o estudo atual do estado-da-arte. Nesse ponto, uma fonte recomendável para obtenção desses recursos é o site *Semantic Scholar*[2] por possuir um banco de mais de 190.000.000 de artigos, permitindo a aquisição de informação relacionada, atual, bem estruturada, de maneira rápida e eficaz.

Além da busca por artigos científicos, o gerenciamento é outra tarefa fundamental, que de forma geral deve ser capaz de organizar as principais informações de cada artigo de tal maneira que facilite compará-lo a outros relacionados ao mesmo assunto. Nesse contexto, há diversos programas que podem realizar esse gerenciamento, tais como: *Zotero, Docear, Mendeley, JabRef, Qiqqa*, entre muitos outros [1].

Apesar desses gerenciadores de artigos facilitarem o processo de organização de bibliografias, eles ainda estão longe de analisar de forma automática o conteúdo semântico dos mesmos. Por esse motivo, hoje há uma demanda crescente por sistemas que possam reconhecer o conteúdo de um artigo científico avançando.

Com a rígida formalização encontrada em artigos acadêmicos, sobretudo científicos, em que divide sua informação entre tópicos, tais como: objetivos, metodologias, revisão bibliográfica, conceitos, experimentos, resultados e discussão, pode-se facilitar as buscas baseadas em Redes Neurais dentro de processos de aprendizado de máquina. Assim, é possível criar modelos matemático-estatístico-computacionais que aprendam a reconhecer essas estruturas formais de artigos de modo a sustentar a construção de arquiteturas de softwares que sejam capazes de organizar essas informações.

Por outro lado, há na literatura uma demanda por agentes rastreadores e organizadores automáticos do conteúdo requerido. A posterior análise automática de artigos baseada em processamento de linguagem natural requer a utilização desse tipo de ferramenta previamente, de modo que possa ser utilizada de forma compartilhada com vários co-autores de projetos.

Assim, este projeto visa o desenvolvimento de uma ferramenta de auxílio à busca automática de artigos científicos em bases de dados como o *Semantic Scholar*, entregando de forma concisa e organizada um conjunto de informações que facilite ao pesquisador a posterior mineração da informação em agentes inteligentes que poderão ser treinados e construídos futuramente utilizando redes neurais [2], visando a extração de informação científica com processamento de linguagem natural [3][4][5].

2. Metodologia

A metodologia proposta neste trabalho é desenvolver um sistema automatizado que receba um termo-chave do usuário e retorna uma tabela com atributos obtidos do site *Semantic Scholar* relacionados a este termo. Além do levantamento bibliográfico do modelo proposto e do objetivo, uma terceira contribuição desse projeto foi a construção de uma base de dados de dados estruturados de artigos científicos. Trata-se de uma base feita no Centro Universitário da FEI, coletada desde 2012 a partir de trabalhos acadêmicos de conclusão de curso, mestrado e doutorado na área de Visão Computacional e Computação Gráfica, todos orientados pelo prof. Paulo Sérgio Rodrigues, que também é orientador deste trabalho. Para que este objetivo seja atingido, um compilador de informações (*crawler*) foi proposto para uma tabela *Excel*. Também foram desenvolvidos dois sistemas de suporte: um responsável por extrair o texto de arquivos .pdf e salva-los em formato .txt para análise futura, e um que será usado para a geração da base de dados, extraindo informações desejadas de tabelas de metadados de artigos que foram preenchidas manualmente.

A metodologia utilizada para a coleta de informação foi a busca manual de artigos na internet, em diferentes bases de dados científicas, mas se destacando a *IEEE, ACM, Springer e Science Direct*. Cada aluno ou equipe de TCC, durante a revisão bibliográfica de seus respectivos trabalhos, executava formalmente uma sequência específica de pesquisa e aquisição de dados em uma tabela.

O resultado final do preenchimento da tabela mencionada acima é uma base de dados de mais de 5.000 (cinco mil) artigos avaliados com as respectivas categorias e informações citadas.

Até onde sabemos, não há disponível na internet uma base de dados com essas características anotadas e tabeladas, e que levaram a mais de 80 trabalhos acadêmicos aprovados em seus diversos níveis de titulação acadêmica.

3. Resultados

A primeira etapa do desenvolvimento deste sistema deu-se como a criação de um *Crawler* desenvolvido em *Python*, utilizando o *framework Selenium*[3], que permite automatizar os cliques em navegadores de internet (neste caso, foi utilizado o *Google Chrome*), e obter metadados das páginas visitadas. Com isso, o termo desejado pelo usuário é inserido no site *Semantic Scholar* e os resultados da pesquisa são salvos com todos os seus metadados disponíveis. A pesquisa é realizada de três maneiras diferentes: de maneira padrão, sem alterar nenhum filtro disponível; utilizando o filtro *Lit Reviews* e pesquisando artigos de no máximo 5 anos de idade. O usuário então escolhe com quantas páginas gostaria que a pesquisa seja realizada, o que resulta em aproximadamente 30 resultados por página pesquisada Após todo o processo de aquisição de metadados, os resultados são salvos utilizando o framework *Pickle*[4] em formato .pkl, de maneira orientada a objetos e de fácil manipulação para processos futuros.

Após todos os metadados terem sido salvos pelo *crawler*, estes precisam ser apresentados de maneira que o usuário possa entender e, caso necessário, editá-los. Para isso, um software também desenvolvido em *Python* foi criado. Usando o framework *xlsxwriter*[5], todos os metadados que foram salvos no formato .pkl pelo *crawler* são lidos e dispostos de maneira que todos os metadados relevantes estejam acessíveis em uma planilha do *Microsoft Excel*. Porém, esses metadados possuem pesos diferentes para cada usuário, uma vez que, para uma pessoa, um artigo mais novo pode ser muito mais relevante que um artigo de 30 anos, porém com milhares de citações. Pensando nisso, foi aplicada à seguinte expressão, Equação (1):

$$P(P_{ot\_fat} \mid P_{vel}, P_{inf}, P_{ano}) = \boldsymbol{\alpha}_1 P_{vel} + \boldsymbol{\alpha}_2 P_{inf} + \boldsymbol{\alpha}_3 P_{ano} \qquad (1)$$

onde $\boldsymbol{\alpha}_1$ é o peso desejado para a velocidade de citação; $\boldsymbol{\alpha}_2$ para o fator de influência; $\boldsymbol{\alpha}_3$ é o fator aplicado à data de publicação do artigo; P_{vel} a velocidade de citação; P_{inf} é o fator de influência e P_{ano} o ano de publicação.

Para melhorar a experiência final do usuário, o *crawler* e o compilador de resultados foram integrados em um único sistema com interface gráfica multi-plataforma (Linux, Windows e MacOS), desenvolvido em *Python* e utilizando um *framework* chamado *appJar*[6], que envelopa a biblioteca *TKinter* [7] (Figura 1).

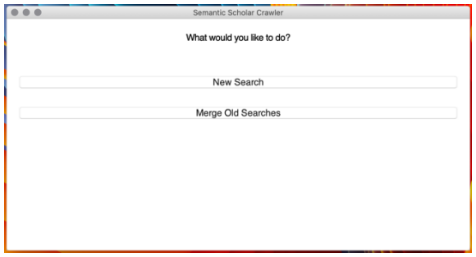

Figura 1 – Tela inicial da GUI

Para que os artigos em .pdf possam futuramente ser interpretados utilizando processamento de linguagem natural, foi desenvolvido um sistema que utiliza o *framework Google Tesseract OCR*[8], que utiliza reconhecimento ótico de caracteres para extrair o texto de um arquivo .pdf. Em seguida, o texto extraído é salvo em formato .txt, o que permite que qualquer outro sistema tenha acesso ao seu conteúdo. Todos os passos são realizados usando multiprocessos, permitindo que a atividade seja dividida para quantas *threads* estejam disponíveis.

4. Conclusões

O presente projeto apresenta um modelo estatístico-computacional para busca e análise de artigos científicos fornecidos no site *Semantic Scholar*. O trabalho possui uma interface em formato de *Crawler* que é capaz de obter, organizar e priorizar metadados de artigos científicos. Como resultado, tem ajudado a revisão bibliográfica de mais de 35 grupos de trabalhos de conclusão de curso do curso de Ciência da Computação da FEI, desde 2017, além de diversos mestrados e doutorados na área de Processamento de Sinais e Imagens. Também foram implementadas todas as ferramentas necessárias de conversão de base de dados, de modo que, nos próximos passos, o modelo possa automaticamente obter informação de análise das principais características dos artigos, tais como: objetivo, metodologias, métricas, base de dados e resultados obtidos.

Link para o projeto: Selenium Semantic Scraper https://github.com/EvertonCa/SeleniumSemanticScraper

5. Referências

[1] José Luis Ortega. Chapter 4 - Reference Management Tools. In Social Network Sites for Scientists, pages 65–99. Chandos Publishing, 2016. DOI: 10.1016/B978-0-08-100592-7.00004-6.

[2] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. ArXiv preprint arXiv:1511.08458, 2015

[3] Steven Bird, Ewan Klein, and Edward Loper. Natural language processing with Python: analyzing text with the natural language toolkit. "O'Reilly Media, Inc.", 2009.

[4] Diego R Amancio. Probing the topological properties of complex networks modeling short written texts. PloS one, 10(2):e0118394, 2015.

[5] Guilherme Alberto Wachs-Lopes and Paulo Sergio Rodrigues. Analyzing natural human language from the point of view of dynamic of a complex network. Expert Systems with Applications, 45:8–22, 2016.

Agradecimentos

À CNPq, CAPES e FAPESP, além do departamento de Ciência da Computação da FEI pela realização das medidas.

[1] Aluno de IC do Centro Universitário FEI. Projeto com vigência de 03/19 a 02/20.

[2] Semantic Scholar url: https://www.semanticscholar.org

[3] Selenium: https://selenium-python.readthedocs.io

[4] Pickle: https://docs.python.org/3/library/pickle.html

[5] xlsxwriter: https://xlsxwriter.readthedocs.io

[6] Appjar: http://appjar.info

[7] TKinter: https://docs.python.org/3/library/tkinter.html

[8] Tesseract: https://opensource.google/projects/tesseract