# ***Intelligent Text***

# **Maikel Maciel Rönnau**

# **Wellington Reis**

**1 Introdução**

Este documento tem o objetivo de apresentar as características técnicas da ferramenta Intelligent Text.

**2 Ferramenta**

Intelligent Text é um editor de texto desenvolvido em Java e Python com o objetivo de facilitar a escrita de textos e artigos técnicos. A ferramenta auxilia o desenvolvimento de textos através da sugestão de artigos relacionados ao assunto sobre o qual o usuário está escrevendo, artigos estes, são buscados no Google Scholar de acordo com palavras-chaves extraídas do texto.

**3 Componentes**

A ferramenta é dividida em dois principais componentes, a interface e o motor de buscas.

**3.1 Interface**

A interface do programa é desenvolvida em Java e dividida entre a área para escrita do texto e outra para a exibição dos artigos encontrados. Na parte de escrita do texto, existe uma função que automaticamente pesquisa por artigos quando o texto tiver no mínimo 200 palavras, após isso, o programa irá buscar artigos toda vez que mais 50 palavras forem adicionadas. O usuário também pode executar uma pesquisa a qualquer momento pressionando o botão *Search* (atente para o número mínimo de palavras necessário).

**3.2 Motor de pesquisa**

O motor de pesquisa é em Python e este é o componente que busca os artigos no Google Scholar. Ele também é dividido em duas partes: a parte que analisa o texto e extrai as palavras-chave; e a parte que utiliza as palavras-chave para buscar artigos relacionados.

**3.2.1 Extração de palavras-chave**

A extração de palavras-chave é feita com o uso das bibliotecas Rake-NTKL e TextRank. A biblioteca Rake-NTKL serve de base para o processamento de linguagem natural feita pela biblioteca TextRank. O resultado do processamento para extração de palavras chave é um *array* destas palavras, que depois são enviadas ao motor de pesquisa propriamente dito.

**3.2.2 Busca de artigos no Google Scholar**

A busca de artigos no Google Scholar utilizando as palavras-chave extraídas do texto acontece com o uso um *script* Christian Kreibich. Seu script executa pesquisas utilizando interface oferecida pelo Google, desta maneira, podendo acessar todos os componentes disponíveis (para executar pesquisas simples até à pesquisas avançadas, com diversos parâmetros). O *script* retorna todos os resultados da primeira página de resultados do Google Scholar. Estes resultados são processados pelo *Intelligent Text* e enviados para a interface.

**3.2.3 Modularização**

O motor de pesquisa foi compilado através do uso da ferramenta PyInstaller, que compila *scripts* Python em executáveis. Desta maneira, é possível executar o motor enviando comandos através do Java. Desta maneira, temos um módulo de interface e outro que é o motor de pesquisa, assim, podemos modificar qualquer um deles sem que isso afete a outra parte, tornando as modificações independentes.

**4 Conclusão**

O desenvolvimento desta ferramenta serviu para o aprendizado dos conceitos de “objeto de aprendizado”, onde utilizamos a tecnologia em favor do ganho de produtividade. Com o uso do Intelligent Text, é possível escrever textos com mais facilidade e também com mais confiabilidade de referencial teórico, já que a ferramenta apresenta artigos relacionados ao texto que está sendo escrito, auxiliando o usuário a escrever com mais fluidez e mais conformidade com a realidade.

# **References**

ADAMO, David. **TextRank**. Repositório Github: https://github.com/davidadamojr/TextRank. Acesso em 24 jun 2017.

KREIBICH,Christian. **Scholar.py.** Repositório Github: https://github.com/ckreibich/scholar.py. Acesso em: 17 jun 2017.

PyInstaller. Website: http://www.pyinstaller.org/.

RÖNNAU, M. Maikel. ***Intelligent Text***. Repositório Github: https://github.com/maikelronnau/intelligent\_text. Acesso em: 27 jun 2017.

 SHARMA, B. Vishwas. **Rake-NLTK**. Repositório Github: https://github.com/csurfer/rake-nltk. Acesso em: 24 jun 2017.