



UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

DEPARTAMENTO DE INFORMÁTICA

SCRIPTING NO PROCESSAMENTO DE LINGUAGEM
NATURAL

Song Lyrics Editor

Grupo 3

Eduardo Gil Ribeiro da Rocha - **A77048**

Manuel Gouveia Carneiro de Sousa - **A78869**

30 de Junho de 2019

Resumo

O presente documento tem como objetivo explicar quais os passos tomados no desenvolvimento de uma ferramenta capaz de construir, para um certo número de palavras, um dicionário de rimas de forma a ajudar o utilizador na criação da letra de uma música.

Conteúdo

1	Introdução	3
2	Trabalho Desenvolvido	3
2.1	Extração de Rimas e Palavras Similares	3
2.2	Ferramenta Produzida	4
2.2.1	Linha de Comandos	4
2.2.2	Interface Web	4
2.3	Exemplificação e Análise de Resultados	5
2.3.1	Linha de Comandos - Língua Portuguesa	5
2.3.2	Interface Web - Língua Inglesa	6
3	Conclusão e Trabalho Futuro	9

1 Introdução

No âmbito da Unidade Curricular de *Scripting no Processamento de Linguagem Natural*, foi-nos proposto o desenvolvimento de diversas ferramentas, sendo uma delas um editor de letras de músicas. A partir de uma lista de palavras fornecidas pelo utilizador, era pretendido que, para cada uma das palavras fossem então extraídas diversas rimas, construindo assim um dicionário composto por estas. Este dicionário seria posteriormente analisado pelo utilizador como forma de auxílio aquando a escrita da letra.

Posto isto, iremos então documentar o processo que levou à criação desta ferramenta, bem como expor as diversas funcionalidades da mesma.

2 Trabalho Desenvolvido

2.1 Extração de Rimas e Palavras Similares

De forma a obter a informação pretendida, o principal foco era o de arranjar serviços dos quais fosse possível a extração de dados para que posteriormente fossem processados. Por essa mesma razão, foi feita uma pesquisa de forma a encontrar *APIs* que fornecessem esse serviço. O intuito inicial era o de permitir que a nossa ferramenta aceitasse palavras tanto da língua portuguesa como da língua inglesa. De facto, não foi possível encontrar uma *API* a qual desse resposta a estes dois pedidos, pelo que foram usadas duas estratégias diferentes.

Em primeiro lugar, para a língua portuguesa, foram então usados *websites* que nos permitiam fazer este tipo de tarefa, mas os quais não tinham uma *API* disponível. Por esta mesma razão, foi então utilizada uma biblioteca do *Python* denominada de *BeautifulSoup*, a qual permite extrair informação de ficheiros *HTML*. Posto isto, foram então usados dois *websites*, *rhymit.com* para o cálculo de rimas e *lexico.pt* para a extração de palavras relacionadas. Em segundo lugar, para a língua inglesa, foi de facto encontrada uma *API* denominada de *datamuse* a qual calcula, para uma palavra dada como *input*, diversos parâmetros, dois deles sendo rimas e palavras similares.

2.2 Ferramenta Produzida

Tendo explicado o processo principal, serão então expostas as funcionalidades que a ferramenta desenvolvida contém, e como poderão ser utilizadas.

2.2.1 Linha de Comandos

Numa primeira fase, foi então desenvolvida uma ferramenta de linha de comandos a qual tratava de construir o dicionário pretendido. Aqui, era fornecida uma lista de palavras passadas como argumento, sendo que posteriormente seriam usadas para obter rimas e palavras similares. Outro dos parâmetros da ferramenta era a língua (português ou inglês), necessário para saber se era usada a *API* ou então o conjunto de *websites* anteriormente mencionados. Por último, era referido o tipo de output pretendido, sendo estes em *JSON*, *HTML* ou *PDF*. No caso de *JSON*, era guardado num ficheiro respetivo o dicionário em bruto. Para *HTML* eram geradas páginas estáticas de forma a permitir uma consulta flexível do dicionário criado. Estas páginas eram geradas a partir de ficheiros *XML* (criados com base no dicionário) para que posteriormente fossem transformados em *HTML* através de *XSLT* (*eXtensible Stylesheet Language for Transformation*). No caso de um *PDF*, é gerado um ficheiro desse mesmo tipo, contendo toda a informação presente no dicionário e uma página referente à escrita da letra da música.

2.2.2 Interface Web

Numa segunda fase, foi construída uma interface *web* de forma a facilitar o uso da ferramenta, alterando ligeiramente a maneira de como eram obtidas as palavras chave. Esta interface é suportada por uma *microframework* escrita em *Python*, denominada de *Flask*. Através desta ferramenta, é possível construir aplicações *web* de forma rápida e flexível.

Relativamente à interface produzida, o processo passa pela escrita de um resumo da música que se pretende escrever, selecionando a língua em que este resumo está escrito (português ou inglês). Posteriormente, são extraídas *keywords* do resumo escrito, podendo eliminar as existentes ou adicionar novas. Por fim, o utilizador tem a possibilidade de gerar um *PDF* contendo o *Worksheet* da música, semelhante ao anterior mas incluindo também o resumo da mesma. De notar que o intuito da interface *web* é apenas de gerar o *PDF*, sendo que os outros formatos de *output* apenas estão disponíveis na ferramenta de linha de comandos.

2.3 Exemplificação e Análise de Resultados

Com o intuito de demonstrar uma possível utilização da ferramenta produzida, decidiu-se então incluir exemplos que contribuíram para uma fase de testes. Aqui, foram incluídos exemplos de uso da ferramenta de linha de comandos bem como da interface *web*.

2.3.1 Linha de Comandos - Língua Portuguesa

A título de exemplo foi incluído um caso em que é escolhida a língua portuguesa, e como output teremos um formato *JSON*. Ao executar o comando, serão então extraídas as páginas *web* relativas às diversas palavras passadas como argumento e, por fim, é construído o dicionário. Executando o programa:

```
./main.py -o json -l pt -w cão gato sapato
```

é, de facto, escrito o dicionário num ficheiro denominado de `suggestions.json`, presente na pasta “out/json/” a partir da diretoria corrente. O dicionário criado terá a respetiva estrutura:

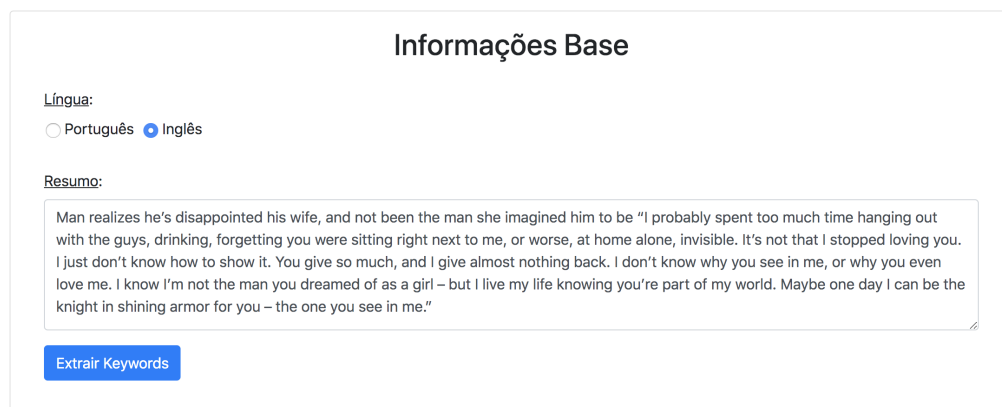
```
{
  "cão": {
    "rhymes": {
      "2": [ ... ],
      "3": [ ... ]
    },
    "similar": [ ... ]
  },
  "gato": { ... },
  "sapato": { ... }
}
```

onde as rimas estarão divididas pelo número de sílabas, e as palavras similares estarão simplesmente dentro de uma lista.

Visto isto, passaremos então a exemplificar o uso da ferramenta tendo em conta uma interface *web*.

2.3.2 Interface Web - Língua Inglesa

Para este exemplo, resolveu-se usar a língua inglesa para a escrita do resumo da música, necessário para extrair as *keywords*.



The screenshot shows a web form titled "Informações Base". It has two main sections: "Língua:" and "Resumo:". Under "Língua:", there are two radio buttons: "Português" (unselected) and "Inglês" (selected). Under "Resumo:", there is a text area containing a paragraph of English text. Below the text area is a blue button labeled "Extrair Keywords".

Informações Base

Língua:

☐ Português ☒ Inglês

Resumo:

Man realizes he's disappointed his wife, and not been the man she imagined him to be "I probably spent too much time hanging out with the guys, drinking, forgetting you were sitting right next to me, or worse, at home alone, invisible. It's not that I stopped loving you. I just don't know how to show it. You give so much, and I give almost nothing back. I don't know why you see in me, or why you even love me. I know I'm not the man you dreamed of as a girl – but I live my life knowing you're part of my world. Maybe one day I can be the knight in shining armor for you – the one you see in me."

Extrair Keywords

Figura 1: Informações Base (Língua e Resumo)

Posteriormente, o utilizador terá que clicar num botão para “Extrair Keywords” do resumo previamente escrito, dando a origem a uma nova secção contendo todas as *keywords* extraídas, e dando a possibilidade de adicionar novas *keywords* ou eliminar as que foram extraídas de forma automática. Para esta extração foi usada uma biblioteca *Python* denominada de *Yake* (Yet Another Keyword Extractor), a qual consegue extrair *keywords* de textos em diversas línguas, sendo esta uma das razões principais para a escolha da mesma.

Keywords	
Adicionar Keyword	Gerar Sugestões
drinking	Eliminar
invisible	Eliminar
man	Eliminar
wife	Eliminar
guys	Eliminar
forgetting	Eliminar
worse	Eliminar
realizes	Eliminar
disappointed	Eliminar
imagined	Eliminar

Figura 2: Keywords extraídas do resumo

Para adicionar uma *keyword* basta clicar no respectivo botão, abrindo assim um *popup* com uma caixa de texto. A palavra escrita será então adicionada à lista de *keywords* existentes.

Adicionar Keyword

Palavra:

Adicionar

Close

Figura 3: Adicionar uma Keyword

Por fim, ao clicar no botão para gerar as sugestões, é então criado o ficheiro *PDF* com as informações base, bem como o dicionário gerado. Este ficheiro é criado do lado do servidor, e posteriormente enviado para o utilizador.

Worksheet

Resumo da música

Man realizes he's disappointed his wife, and not been the man she imagined him to be "I probably spent too much time hanging out with the guys, drinking, forgetting you were sitting right next to me, or worse, at home alone, invisible. It's not that I stopped loving you. I just don't know how to show it. You give so much, and I give almost nothing back. I don't know why you see in me, or why you even love me. I know I'm not the man you dreamed of as a girl – but I live my life knowing you're part of my world. Maybe one day I can be the knight in shining armor for you – the one you see in me."

Keywords

1. forgetting
2. worse
3. realizes
4. disappointed

Figura 4: Excerto do Worksheet gerado

3 Conclusão e Trabalho Futuro

Com o trabalho realizado, e visto que este consistiu na construção da ferramenta através do auxílio da linguagem *Python*, o grupo sente uma forte melhoria nas técnicas e abordagens usadas na resolução de problemas. De facto, as diversas funcionalidades implementadas fizeram com que todo o conhecimento sobre a linguagem fosse aprimorado.

Quanto à solução produzida, o grupo sente que os objetivos foram atingidos. Para além de uma ferramenta de linha de comandos capaz de construir um dicionário de rimas e o mostrar em diversos formatos, foi ainda criada uma aplicação *web* a qual traz a mesma funcionalidade, mas facilitando o seu uso para qualquer utilizador comum.

Como trabalho futuro, considera-se que o suporte a diversas línguas seria o mais importante, dando assim a possibilidade de utilização a um maior número de pessoas.

Referências

- [1] **Datamuse** - <https://www.datamuse.com/api/>
- [2] **Rhymit** - <http://www.rhymit.com/>
- [3] **Lexico** - <https://www.lexico.pt/>
- [4] **Beautiful Soup** - <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- [5] **Jinja2** - <http://jinja.pocoo.org/docs/2.10/>
- [6] **RAKE** - <https://pypi.org/project/rake-nltk/>
- [7] **Yake** - <https://github.com/LIAAD/yake>
- [8] **Song Chops** - <http://songchops.com/>