Trabalho 1 – Terrier

Recuperação da Informação – 2019.1
Universidade Federal do Rio de Janeiro

Aluno: Thiago Henrique Neves Coelho – 116016400

1. Introdução e Objetivo

Terrier é um sistema de Recuperação da Informação *open source* escrito em Java. É desenvolvido pela School of Computing Science, University of Glasgow.

O objetivo deste trabalho é relatar a utilização do Terrier para indexação e realização de consultas em uma base de documentos, além de dar uma visão geral sobre as funcionalidades do sistema e como instalá-lo.

2. Instalação

O sistema operacional utilizado foi o Windows 10 e a versão do Terrier foi a v5.1.

Primeiramente, deve-se baixar o sistema em http://terrier.org/download/. A versão utilizada corresponde a “terrier-project-5.1-bin.zip (binaries)”. Após isso, extraia o conteúdo para algum lugar.

Por fim, é necessário fazer algumas alterações nas variáveis de ambiente:

- O caminho para o java deve estar no Path, ou seja, deve ser possível chamar o comando “java” no terminal diretamente;
- Crie uma variável JAVA_HOME com o valor do caminho para a pasta do Java JRE ou JDK;
- Crie uma variável TERRIER_HOME com o valor do caminho para a pasta que foi extraída anteriormente.

OBS: É necessário que os valores de JAVA_HOME e TERRIER_HOME não possuam espaços em branco. No Windows 10, é possível substituir os espaços em branco por %20. Exemplo: C:\Program%20Files\Java\jre1.8.0_171.

3. Pré-processamento dos Documentos

O Terrier possui suporte para línguas além do inglês. A tokenização e o *encoding* padrões levam em consideração o inglês como idioma, mas essas configurações podem ser mudadas em um arquivo chamado *terrier.properties,* que será explicado no próximo tópico. Entretanto, consultas com palavras acentuadas não funcionaram após as tentativas de ajustar as configurações.

A solução encontrada para isso foi criar um programa em python para remover todos os caracteres especiais dos documentos e da lista de *stopwords*. Este script foi disponibilizado no seguinte ink:
https://github.com/Thiagohnc/rec-info/blob/master/remove_char_especial.py

4. Configurando o Terrier e Indexando Documentos

Primeiramente devemos informar quais são os documentos a serem indexados. Para isso, navegue pelo terminal até a pasta extraída anteriormente (terrier-project-5.1) e digite o seguinte comando:
bin\trec_setup.bat <caminho-para-a-pasta-com-os-documentos>

Dentro da pasta do Terrier, entre na pasta "etc" e será possível perceber que alguns arquivos de propriedades foram criados. Em *collection.spec* é possível ver todos os arquivos que estavam na pasta passada como caminho para o comando *trec_setup.bat*, é possível editar *collection.spec* e retirar os documentos que não devem ser indexados, por exemplo um arquivo README.

Outro arquivo importante nesta pasta é o *terrier.properties*. Ao abri-lo em um editor de texto, pode-se perceber que há diversas configurações definidas nele. É possível mudar não apenas essas, como muitas outras configurações que não estão no arquivo mas possuem algum valor padrão. Algumas mudanças úteis neste arquivo são:

- ***Stemmer***
  - Alterando a linha *termpipelines=Stopwords,PorterStemmer* para *termpipelines=Stopwords,PortugueseSnowballStemmer*, definimos que o *stemmer* utilizado será para palavras em português
- ***Stopwords***
  - A lista de *stopwords* padrão é baseada no vocabulário em inglês, mas é possível utilizar uma lista de *stopwords* customizada definindo o valor de *stopwords.filename* como o caminho para a lista de *stopwords*. Esse *filename* será utilizado dentro do programa em Java, então barras invertidas devem ser dobradas. Exemplo: *stopwords.filename=C:\\Users\\thiag\\Documents\\stopwords_pt.txt*. A lista de *stopwords* em português utilizada para este trabalho foi a do seguinte link: https://gist.github.com/alopes/5358189#file-stopwords-txt.
- **Método de Ponderação de Termos**
  - Também pode ser definido o método de ponderação dos termos utilizado. O padrão utilizado pelo Terrier é o DPH, que é descrito como "A different hyper-geometric DFR (Divergence from Randomness) model using Popper's normalization (parameter free)". Há várias outras opções de modelos, entre eles: BM25, TF-IDF, Hiemstra_LM, etc.
  - Para alterar o método utilizado, devemos alterar o arquivo *terrier.properties*, definindo o valor de *trec.model* para o método desejado. Por exemplo, caso seja desejado utilizar o método BM25, deve-se adicionar a linha *trec.model=BM25* no arquivo.

Após as configurações terem sido definidas, deve-se fazer a indexação dos documentos utilizando, na pasta do Terrier, o comando *bin\terrier batchindexing*.

5. Realizando Consultas

Deve-se criar um arquivo xml para as consultas a serem feitas. É possível fazer várias consultas, numerando cada uma. Um exemplo de xml de consulta é o seguinte:

```
<top>
<num> 1 </num>
<PT-title> consulta </PT-title>
</top>
```

Para realizar a consulta, deve-se executar o seguinte comando:
bin\terrier batchretrieval -t <caminho-para-o-arquivo-com-a-consulta>

Dentro da pasta do Terrier, há uma pasta chamada “var” e dentro dela há uma chamada “results”. Nesta pasta haverá um arquivo com extensão “.res”, nele está o resultado da consulta realizada.

Podemos utilizar uma linguagem específica para os termos da consulta, possivelmente obtendo um resultado melhor com isso. Algumas formas de formular uma consulta são:

- termo1 termo2
  - Retorna documentos que contêm o termo1 ou o termo2
- +termo1 +termo2
  - Retorna documentos que contêm o termo1 e o termo2
- +termo1 -termo2
  - Retorna documentos que contêm o termo1 e não contêm o termo2
- termo1^2.3
  - Multiplica o peso do termo1 por 2,3
- “termo1 termo2” ou +“termo1 termo1”
  - Retorna documentos em que o termo1 e o termo2 aparecem em uma frase

Consultas interativas

Um modo mais rápido de realizar consultas com o Terrier é com o comando *bin\terrier interactive*. Dessa forma, é possível digitar a consulta e receber os resultados direto no terminal de forma interativa, sem necessitar do uso de arquivos.

Para a consulta interativa, apenas a consulta deve ser escrita, logo os termos não estarão em um arquivo nem devem estar no formato que a consulta teria caso estivesse em um arquivo xml, ou seja, não deve-se colocar <top>, <num>, etc.

6. Consultas na Coleção de Teste

Para realizar as consultas, foi criado um arquivo xml com as 10 consultas enumeradas (com o número no campo <num></num>. Foram utilizadas as técnicas de stemming e remoção de stopwords descritas anteriormente. Além disso, as consultas foram formuladas usando as técnicas da linguagem de consultas do Terrier. A formulação de cada consulta foi a seguinte:

- Consulta 1: Boicotes de consumidores
  - A consulta foi +boicotes +consumidores^1.5 -politico. Isso foi feito para pegar documentos que falem sobre boicotes e sobre consumidores e não falem sobre “politico”, pois boicotes políticos não são relevantes para a consulta. Além disso, para garantir maior chance de ser um boicote de consumidores, foi atribuído um peso maior para “consumidores”
- Consulta 2: Atividades do grupo terrorista ETA na França
  - A consulta foi *+Franca +terrorista^2 +ETA*. A intenção dessa formulação foi retornar documentos que falam sobre a França, terroristas e ETA. Para gerar resultados menos genéricos, foi atribuído peso 2 à palavra “terrorista”
- Consulta 3: Filmes passados na Escócia
  - A consulta foi *+filme +"passa Escocia"*. A intenção da formulação foi encontrar documentos que falem de filme e contenham “passa” e “Escócia” em uma frase. A motivação disso foi tentar encontrar documentos com frases do tipo “… se passa na Escócia… ”, etc.
- Consulta 4: Tratamento de resíduos industriais
  - A consulta foi +tratamento +residuos +industria. A formulação foi bem simples e apenas procura por documentos que tenham os três termos
- Consulta 5: Desemprego na Europa
  - A consulta foi +desemprego +Europa +porcentagem. Essa formulação teve como objetivo retornar documentos que falem sobre “desemprego”, “Europa” e “porcentagem”. A intenção de incluir “porcentagem” foi encontrar documentos que deem dados estatísticos e números sobre o problema

- Consulta 6: Celebrações de centenários
  - A consulta foi +centenario +celebracao
- Consulta 7: Espécies em via de extinção
  - A consulta foi +especie +extincao +Europa
- Consulta 8: Greves
  - A consulta foi +greve^2 +motivo +razao. A intenção dessa formulação foi retornar documentos que falem sobre motivos/razões de greves. Para encontrar documentos mais específicos sobre greves, foi atribuído um peso maior para o termo “greve”
- Consulta 9: Produção global de ópio
  - A consulta foi *opio^3 papoila^3 papoula^3 cultivo producao*. As palavras mais importantes da consulta são “opio”, “papoila” e “papoula”, por isso foi atribuído um peso grande a elas. Papoula é sinônimo de papoila
- Consulta 10: Crises energéticas
  - A consulta foi falta energia^1.5 combustivel^1.5. A intenção foi rankear documentos utilizando esses 3 termos. As palavras mais relevantes eram “energia” e “combustivel”, por isso receberam pesos maiores.

7. Integração com Aplicação Java

Uma funcionalidade do Terrier que não foi explorada no trabalho, mas que vale ser citada é a possibilidade de integrar o sistema com uma aplicação Java. Segundo o site oficial do Terrier, um dos casos de uso comuns para a ferramenta é como um componente de busca dentro de uma aplicação grande.

Dessa forma, é possível fazer indexação e recuperação de documentos na aplicação utilizando o Terrier.

8. Referências

- http://terrier.org