I-Preproc: uma ferramenta para pré-processamento e indexação incremental de documentos

Renan Gomes Pereira[1]
Maria Fernanda Moura[2]

Resumo: O objetivo deste trabalho é apresentar a ferramenta I-Preproc, utilizada no pré-processamento e indexação incremental de documentos. A ferramenta foi implementada em Java utilizando a biblioteca open source Apache Lucene. Embora a ferramenta esteja em desenvolvimento, os resultados parciais obtidos têm sido bastante satisfatórios, como mostrado no experimento realizado.

Palavras-chave: mineração de textos, Apache Lucene, máquina de busca.

Introdução

I-Preproc é uma ferramenta flexível e expansível desenvolvida em Java utilizando a biblioteca open source Apache Lucene (APACHE SOFTWARE FOUNDATION, 2015), para pré-processamento, indexação incremental e busca de documentos. O usuário especifica os parâmetros da indexação por meio de um arquivo de configuração em XML, no qual podem ser especificados quais filtros serão aplicados nos documentos na fase de indexação e, caso o documento seja um arquivo XML, as tags do documento que terão seus conteúdos indexados. Além disso, é possível informar os parâmetros desejados durante a extração dos resultados na forma de matrizes atributo-valor. Essas matrizes são utilizadas em processos de mineração de textos,

[1] Engenharia da Computação, Universidade Estadual de Campinas; Estagiário da Embrapa Informática Agropecuária, Campinas, SP.

[2] Estatística, doutora em Ciências Matemáticas e da Computação, pesquisadora da Embrapa Informática Agropecuária, Campinas, SP.

por ferramentas de aprendizado de máquina. Nessas matrizes, as linhas correspondem aos documentos (observações, instâncias), as colunas aos atributos (palavras, composições de palavras, frases, etc.) e cada célula ao grau de importância do atributo para o documento correspondente; por exemplo, o valor da célula pode representar a frequência de uma palavra em um texto.

A possibilidade de realizar a indexação incremental de uma coleção extensa documentos é crucial. Neste tipo de indexação, a ferramenta insere novos documentos em um índice já existente sem a necessidade de reindexar toda a coleção. Esse processo economiza tempo e recursos computacionais que seriam gastos na reindexação de documentos que já estavam indexados.

Por fim, com a I-Preproc também é possível fazer buscas nos índices por palavras, N-gramas, frases, termos exatos, e, também excluir termos dos resultados. Caso o número de documentos retornados na busca seja menor que um valor determinado pelo usuário, a ferramenta realiza uma outra busca por termos similares utilizando o conceito de edit distance (Konchady, 2008).

Dessa forma, a I-Preproc é uma ferramenta flexível para ser usada em indexação, busca e geradora de dados para ferramentas de aprendizado de máquina, que pode e está sendo evoluída (e desenvolvida) de forma incremental, com uma boa performance, conforme apresentado neste trabalho.

Materiais e Métodos

Nesta seção é mostrado o experimento de performance realizado e um exemplo de um arquivo de configuração para a indexação de uma base de textos, bem como a arquitetura de alto nível da I-Preproc, com a explicação da função de cada módulo constituinte.

- **Experimento**: foi realizado um primeiro experimento para avaliar o tempo de indexação, com uma base exemplo de 2054 textos de tamanhos variáveis. As opções para a indexação, nesse experimento, foram: a) utilizar um vocabulário controlado (o disponível para testes contém apenas 33178 unigramas); b) indexar e filtrar o texto completo removendo acentuação, convertendo letras maiúsculas para minúsculas e removendo caracteres

especiais. Foram indexados apenas os termos stemmizados; quanto ao idioma, o escolhido foi o português do Brasil, e os testes foram feitos em uma máquina Intel(R) Core(TM) i3-2120 CPU @ 3.30GHz com 8GB de RAM rodando no sistema operacional Ubuntu 14.04.

- **Arquivo de configuração**: neste arquivo é possível fornecer todos os parâmetros desejados na indexação, na extração das matrizes atributo-valor e na busca. Um exemplo de arquivo de configuração para a indexação é mostrado na Figura 1, onde:

```
<IncPreProc>
  <Paths indexdir = "../indexdir/"
         textbase = "../textbase/"
         incrementaltextbase = "../textbase/newtexts/"
         domainVoc = "../vocabularies/domainvoc.txt"
         stoplist = "../stopwords_bra.txt"/>
  <Tags
         xmlTags = "author, location, content"
         filteredXmlTags = "content" />
  <Ngram ngrams = "false" maxNGrams = "4"/>
  <Stemmization language = "BRA" stemming = "STEM_NO_STEM"/>
  <Filters domainVoc = "true"
         stoplist = "false"
         toLowercase = "true"
         removeAccents = "true"
         removeSpecials = "true"/>
</IncPreProc>
```

Figura 1. Exemplo de um arquivo de configuração.

<Paths>: especifica os endereços dos arquivos e diretórios necessários para a indexação.

<Tags>: quais serão as tags de um arquivo XML a serem indexadas e quais dessas tags serão filtradas. Caso <Tags> seja removida do arquivo, a ferramenta indexa e filtra o arquivo de texto completo.

<Ngrams>: opções para a indexação de N-gramas, se serão indexados e qual o tamanho máximo de cada n-grama.

<Stemmization>: escolher dentre as opções de indexação. As opções podem ser: a) indexar apenas as palavras sem stemmizar; b) indexar apenas as palavras stemmizadas; c) indexar ambas as formas. Também

é fornecido o idioma de stemmização. As opções disponíveis no momento são inglês “ENG” e português brasileiro “BRA”. Pretende-se incluir novos idiomas como o francês e o espanhol.

<Filters>: são escolhidos os filtros que se deseja usar. Os filtros disponíveis até o momento, estão listados abaixo:

- Vocabulário controlado: especifica-se o arquivo que contém um vocabulário controlado. O vocabulário controlado é escolhido pelo usuário.
- Stopwords: se o usuário optar por não usar um vocabulário controlado, é possível fornecer um arquivo com stopwords que são palavras que o índice deve desconsiderar. As stopwords são escolhidas e especificadas pelo usuário.
- Indexar em letras minúsculas: desconsidera a diferença entre letras maiúsculas e minúsculas.
- Remover acentuação: remove os acentos das palavras dos textos antes de indexar.
- Remover caracteres especiais: remove todos os caracteres diferentes de letra, número e hífen.

Arquitetura da I-Preproc: a arquitetura de alto nível do projeto está presente na Figura 2. A I-Preproc é composta por 4 módulos e um índice criado pela biblioteca Lucene:

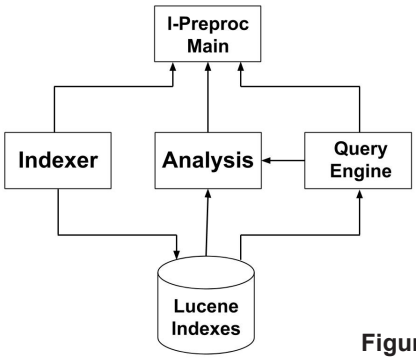

Figura 2. Arquitetura da I-Preproc.

I-Preproc Main é o módulo responsável pela comunicação entre os módulos da ferramenta e pelo controle do fluxo de execução do programa. A comunicação é feita por meio de um arquivo de comunicação que é salvo quando um índice é criado. Este arquivo contém todos os parâmetros que foram usa-

dos para criar este índice, como os filtros utilizados, tamanho dos N-gramas e opções de stemmização. Quando algum módulo opera sobre este índice, ele carrega o arquivo de comunicação deste índice. Deste modo, é possível manter a coerência do índice e das operações efetuadas nele.

O **Módulo Indexer** realiza a indexação incremental de novos textos sem a necessidade de reindexar toda a coleção. A I-Preproc cria um índice de acordo com as opções escolhidas pelo usuário no arquivo de configuração. Após a criação desse índice, quando o usuário deseja indexar novos textos, ele fornece o diretório onde estão os novos textos (este diretório deve ser diferente do diretório da coleção inicial). A ferramenta carrega o arquivo de comunicação, que contém as informações do índice, e indexa todos os textos contidos neste novo diretório movendo-os para o diretório da coleção total. Como, em geral, o número de textos a serem adicionados incrementalmente no índice é relativamente bem menor do que a coleção total, esse processo é muito mais rápido do que a reindexação da coleção completa.

O **Módulo Analysis** extrai as matrizes atributo-valor que contém informações estatísticas importantes sobre o índice e tem formato adequado para ferramentas de aprendizado de máquina. O usuário pode escolher, pelo arquivo de configuração, quais as medidas que ele deseja extrair, como TF ou TF-IDF. Também é possível filtrar os resultados a serem colocados na matriz utilizando filtros de frequência.

O **Módulo Query Engine** é responsável pelas buscas no índice. As buscas podem ser feitas por palavras, N-gramas, frases, termos exatos, e, também excluir termos dos resultados. Caso o número de documentos retornados na busca seja menor do que um valor determinado pelo usuário, a ferramenta realiza uma outra busca por termos similares utilizando o conceito de edit distance.

Resultados e Discussão

Os intervalos em segundos para 10 execuções do programa se encontram na Tabela 1. Ao indexar incrementalmente 78 textos em temas agrícolas de tamanhos variados no índice descrito acima, o tempo médio em 10 execuções foi de aproximadamente 2 segundos (2,183 segundos).

Tabela 1.

Media	10,8125
Desvio padrão	0,1849133431
Intervalo de confiança	0,1146083714
Intervalo inferior	10,697891629
Intervalo superior	10,927108371

Tabela 2.

Termos processados	7153
Media	20,2748
Desvio padrão	0,1742455483
Intervalo de confiança	0,1079965253
Intervalo inferior	20,166803475
Intervalo superior	20,382796525

Na Tabela 2 são mostrados os tempos obtidos para 10 execuções do módulo Analysis sobre o índice criado acima para a extração de uma matriz atributo-valor utilizando a medida estatística TF-IDF.

Foram feitas algumas buscas no índice criado com termos que estão presentes no vocabulário controlado e com alguns termos que não estão no vocabulário controlado. Como esperado, os termos não presentes no vocabulário controlado não foram retornados como resultado de uma busca, pois não foram indexados. Para os termos pesquisados que estavam presentes no vocabulário controlado, alguns exemplos de resultados estão ilustrados na Figura 3.

Como ilustrado na Figura 3, a busca encontrou rapidamente os termos que estavam presentes no vocabulário controlado. O módulo Query Engine aplica os mesmos filtros que o Indexer aplicou na fase de criação do índice e neste caso é indiferente pesquisar com acentos, ou letras maiúsculas ou

```
Insira a busca: leite
Sua busca retornou 205 resultados.
Tempo: 2 ms
Insira a busca: abacaxi
Sua busca retornou 15 resultados.
Tempo: 2 ms
Insira a busca: ABÁCÂXÍ
Sua busca retornou 15 resultados.
Tempo: 1 ms
Insira a busca: Irrigação
Sua busca retornou 0 resultados.
Tempo: 12 ms
```

Figura 3. Resultado da busca no índice.

minúsculas. Quando a busca não encontra resultados, o tempo é um pouco maior pois ela tenta pesquisar termos similares utilizando o conceito de edit distance.

Considerações Finais

Com uma coleção relativamente grande de textos, os tempos de execução para uma única máquina com capacidade de processamento de um computador residencial atual foram bem baixos. Como trabalhos futuros, serão implementadas novas funcionalidades, tais como, o tratamento de sinônimos e relações taxonômicas entre termos para indexação e busca, bem como novas formas de extração da matriz atributo-valor, tais como agrupamentos de atributos por classes pré-determinadas e agrupamentos de documentos, por exemplo, por publicações referentes a um mesmo tema. Além disso, pretende-se melhorar o planejamento experimental para avaliar a performance da ferramenta.

Referências

APACHE SOFTWARE FOUNDATION. **Apache Lucene Core**. Disponível em: <https://lucene.apache.org/core/>. Acesso em: out. 2015.

KONCHADY, M. **Building search applications**: Lucene, LingPipe, and Gate. Oakton: Mustru Pub., 2008. 430 p. ill.