

Tiger2XCES

Mírian Bruckschen^{1,2}, José Guilherme Camargo de Souza^{1,2}, Renata Vieira¹

¹Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Av. Ipiranga 6681 – 90619-900 – Prédio 32 (FACIN) – Porto Alegre – RS – Brasil

²Universidade do Vale do Rio dos Sinos (UNISINOS)
Av. Unisinos 950 – 93022-000 – São Leopoldo – RS – Brasil

{mirian.bruckschen, joseguilhermecs, renata.vieira}@gmail.com

1. Sobre

Este documento é um LEIAME do conversor Tiger2XCES.

2. Para rodar o programa

A partir dos fontes:

No diretório raiz (*tiger2xc.es*), digitar:

```
java -classpath .:ecs.jar core/Tiger2XCES <arquivo.tiger.xml>
```

O programa irá gerar 3 arquivos de saída: um de tokens, um de pos e outro de phrases. Por padrão os arquivos de saída serão criados no mesmo diretório do arquivo de entrada. Por exemplo: se o arquivo de entrada especificado for */tempdir/arquivotiger.xml*, os 3 arquivos de saída serão criados em */tempdir*.

Caso seja desejado gravar os arquivos de saída em outro diretório, pode-se usar a opção *-d* na linha de comando. Exemplo:

```
java -classpath .:ecs.jar core/Tiger2XCES -d /corpus_annotado texto1.xml
```

onde *-d* é a opção para especificar o diretório de saída e */corpus_annotado* o diretório especificado.

A partir do binário (*tiger2xc.es.jar*):

```
java -jar tiger2xc.es.jar <arquivo.tiger.xml>
```

para converter um arquivo. Assim como no primeiro exemplo (acima), são gerados três arquivos. Para especificar um diretório que não o mesmo diretório do arquivo de entrada:

```
java -jar tiger2xc.es.jar -d <diretório_de_saída> <arquivo.tiger.xml>
```

Em ambos os exemplos acima é necessário que o arquivo *ecs.jar* esteja no mesmo diretório que o arquivo *tiger2xc.es.jar*. Caso contrário um erro como este aparecerá:

```
zk@box ~/tiger2xc.es/tiger2xc.es $ java -jar tiger2xc.es.jar ~/pln/pln-br/foo.xml
createPhrase: s2_507 has not edges.
Check PALAVRAS Tiger-XML output for the edges.
If there are no edges in this nt, this could be a PALAVRAS bug.
Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/ecs/Element
    at core.TigerConversor.generateTokenFile(TigerConversor.java:740)
    at core.TigerConversor.generateXCESFiles(TigerConversor.java:695)
    at core.TigerConversor.processFile(TigerConversor.java:153)
    at core.Tiger2XCES.main(Tiger2XCES.java:35)
```

Outro ponto importante a ressaltar é que o arquivo Tiger-XML de entrada deve ser gerado pela versão mais recente do PALAVRAS, chamada pelo Eckhard Bick de PALAVRAS 2. Arquivos antigos, de antes do segundo semestre de 2006 provavelmente falharão na conversão.

3. Execução para múltiplos arquivos

Para rodar o Tiger2XCES para múltiplos arquivos, deve-se utilizar um script auxiliar, *executar_tiger2xc.es.sh*, anexo. Este deve ser adaptado para a execução na máquina em que for utilizado. A linha 6 deve ser alterada da seguinte forma:

```
java -jar <dir-do-tiger2xc.es>/tiger2xc.es.jar -d <dir-destino-dos-arquivos-xc.es> $1
```

O script deve ser executado da seguinte forma:

```
$ ./executar_tiger2xc.es <dir-com-arquivos-xml>/*.xml
```

4. Bugs conhecidos

- Existe um bug conhecido relativo ao escape de caracteres ampersand (&). Caso esse problema aparece é necessário criar um script para converter todos os caracteres & para & a fim de evitar problemas de parsing em browsers e bibliotecas.
- Arquivos Tiger-XML muito grandes podem causar um estouro de memória.

5. Limitações

- Interface. Não existe uma interface gráfica. Somente por linha de comando.
- Os arquivos XML gerados não têm espaços entre os elementos. A biblioteca utilizada (ECS) para gerá-los não possui a opção de indentação de código. Caso seja necessário indentar os arquivos pode-se usar o programa xmllint, presente na maioria das distribuições Linux. A sintaxe é descrita abaixo. Se não for fornecido a segunda parte do comando, isto é, > *nomedoarquivo_indentado.xml*, a saída do programa será direcionada pra tela. Uma idéia é fazer um script (bat para Windows ou Bash para Linux) para aplicar esse programa toda vez que o Tiger2XCES é executado.

```
xmllint --format nomedoarquivo.xml > nomedoarquivo_indentado.xml
```

6. Pré-processamento

Antes de executar o PALAVRAS, cabe lembrar que a codificação de caracteres aceita é ISO-8859-1. Atualmente, alguns textos vem na codificação UTF-8. Se este for o caso, pode-se executar o script *utf2latin*, anexo a este pacote. Ele requer Python instalado, apenas. Modo de usar:

```
$ ./utf2latin <input_directory> <output_directory> [original_encoding]
```

O PALAVRAS gera o XML do Tiger-XML mal-formado (not well-formed). Os valores de alguns atributos apresentam valores com os caracteres < e >. Ambos não são permitidos como valores em arquivos XML e portanto devem ser escapados, respectivamente, para < e >. Para que o Tiger2Xces processe corretamente os arquivo

Tiger-XML gerados pelo PALAVRAS, esses arquivos devem ser pré-processados. Alguns exemplos de más-formações presentes nos arquivos Tiger-XML gerados pelo PALAVRAS são “<FOC”, “NUM>”, “<fmc>”, “AsS<”, “<NER” e a tag “lixo”. Para corrigir este problema, foi criado o script *remove_error_tags.sh*, anexo, que trata de várias destas questões. Entretanto, é possível que alguns erros permaneçam e precisem de tratamento manual, por não ser possível corrigi-los automaticamente. Um exemplo é quando fala-se em minutos e segundos na forma de aspas e apóstrofes (ex.: 10’59”). Para executar o script, basta usar o seguinte comando:

```
$ ./remove_error_tags <dir-com-arquivos-xml>/*.xml
```

Depois da execução do script *remove_error_tags.sh*, os arquivos XML terão a extensão *modified*. Para renomear todos ao mesmo tempo, pode-se executar o seguinte comando no diretório onde estão os arquivos:

```
$ rename 's/\.xml.modified/\.xml/' *.modified
```
