vnTokenizer 4.1.1 Userguide

28, December 2009

1 Introduction

vnTokenizer is a software for tokenizing Vietnamese texts. It segments Vietnamese texts into lexical units (words, names, dates, numbers and other regular expressions) with a high accuracy, of about 98% on a test set extracted from the Vietnamese treebank.

vnTokenizer is written in Java. The software requires Java Runtime Environment 1.6+ to be installed. **vnTokenizer** is distributed under GNU/GPL license.

2 Quickstart

The software is designed to be used from the command line or programmatically via its API. To run the program, use the appropriate script for your operating system: `vnTokenizer.bat` (MS Windows) or `./vnTokenizer.sh` (Unix, Linux, MacOS X).

In a run **vnTokenizer** can tokenize a text file or a set of text files (contained in a directory). Input files must be plain text files encoded in UTF-8 encoding. Results are saved to plain text files or simple XML files and are always encoded in UTF-8 encoding.

2.1 Tokenize a file

To tokenize a file:

```
./vnTokenizer.sh -i <inputFile> -o <outputFile> [options]
```

The allowed options are:

-xo (xml output) – write result to a simple XML file instead of the default plain text format;

-nu (no underscore) – don't connect syllables of a word by an underscore character, space will be used between syllables of composed words instead of the default underscore character;

-sd (sentence detection) – detect sentences before tokenizing. If this option is used, **vnTokenizer** first detects sentences of the input file and then tokenizes the detected sentences. By default, **vnTokenizer** processes the whole text without splitting it into sentences. Therefore, if the text is long, this option should be used to speed up the tokenization.

These options can be used together to produce desired result. Examples:

- `./vnTokenizer.sh -i samples/0.txt -o samples/0.tok.txt`
- `./vnTokenizer.sh -i samples/0.txt -o samples/0.tok.xml -xo`
- `./vnTokenizer.sh -i samples/0.txt -o samples/0.tok.txt -sd`

2.2 Tokenize a directory

To tokenize all files in a directory:

```
./vnTokenizer.sh -i <inputDirectory> -o <outputDirectory> [options]
```

The `inputDirectory` contains files which need to be tokenized. The `outputDirectory` is an empty directory for storing result. Note that these directories must exist in your system. The options are similar to those above, plus one option

-e (extension) – The extension of the files in the input directory. By default, **vnTokenizer** scans the input directory for `.txt` files to tokenize. If you want to tokenize `.xyz` files, use the options `-e .xyz`

Thus, the command:

```
./vnTokenizer.sh -i input -o output -e .xyz
```

will process all files `input/*.xyz` and write the result to `output/*.xyz`. Note that each result file has the same name with that of the corresponding input file.

3 Resources and API

3.1 Resources

All required resources of **vnTokenizer** are stored in directory `models/tokenization`. If you want to integrate `vnTokenizer` into your program, you should copy this directory (along with all of its subdirectories) to your system.

3.2 API

The main class of **vnTokenizer** is `vn.hus.nlp.tokenizer.VietTokenizer`. This class provides some public methods for tokenizing texts and producing results:

1. `public void tokenize(String inputFile, String outputFile)`
2. `public String[] tokenize(String text)`
3. `public void tokenizeDirectory(String inputDir, String outputDir)`
4. `public String segment(String sentence)`

The following code snippet creates and runs a Vietnamese tokenizer on a file:

```
VietTokenizer tokenizer = new VietTokenizer();
tokenizer.tokenize(inputFile, outputFile);
```

You may also want to look at class `vn.hus.nlp.tokenizer.Tokenizer`. This class provides a method for tokenizing a `Reader` and a method for getting result:

1. `public synchronized void tokenize(Reader reader) throws IOException`
2. `public List<TaggedWord> getResult()`

You can also develop your own `Outputer` for controling the output format of the result (classes `Outputer`, `IOutputFormatter`).

3.3 Tokenizing service

vnTokenizer may also be run as a service for other systems (that are not nesscessary developed in Java) to connect to, tokenize a stream of data and get the tokenization result. For more information and usage of this service, please take a look at the package `vn.hus.nlp.tokenizer.web`.

4 Contents

File/Directory	Description
vn.hus.nlp.tokenizer-*.jar	The JAR file containing all classes
README-vi.txt	A readme file (in Vietnamese)
README-en.pdf	Userguide (in English)
resources	Required resources and data
lib	Required libraries
samples	Some samples

If you want to add new words to the lexicon of **vnTokenizer**, simply add them to `models/tokenization/automata/externalLexicon.xml`.

5 Contact

For more information, bug reports, fixes, contact: Le Hong Phuong

- Address: Department of Mathematics, Mechanics and Informatics, Vietnam National University, Hanoi, Vietnam
- Email: *phuonglh@gmail.com*

6 References

- Website of **vnTokenizer**: *http://www.loria.fr/∼lehong/tools/vnTokenizer.php*
- "A hybrid approach to word segmentation of Vietnamese texts". L. H. Phuong, N. T. M. Huyen, R. Azim, H. T. Vinh. *Proceedings of the 2nd International Conference on Language and Automata Theory and Applications*, LATA 2008, Springer LNCS 5196, Tarragona, Spain, 2008.