kemangi Documentation

Release 1.0.0

William Gozali

October 05, 2015

Contents

Welcome to Kemangi!

Here is what might help you:

CHAPTER 1

Introduction

1.1 Overview

Kemangi is Indonesian Language (Bahasa Indonesia) pre-processor tool.

If you are going to conduct text mining related research with Indonesian Language, there is a high probability that pre-processing is needed. The purpose of Kemangi is to clean up your text file so that it is easier to be processed.

Kemangi provides basic task pre-processing task:

- Remove non-ASCII characters
- Remove non-alphanumeric characters
- Case folding
- Remove stop words (meaningless word)
- Remove words according to pattern provided by you, e.g. hashtags, URLs, Twitter mentions, etc
- Word stemming

1.2 Example

Raw input:

```
Mempermainkan peranan 12 domba di pementasan
ALAyISME iTu TETAP ada di Jakarta
Saya tidur... kemarin
Pin  BB saya   B12A3FC
bbm koq naik, warga sedih #edisicurhat
dia memblokir website http://www.lucu.com
éà ada karakter ga jelas, non ASCII appeared! #wow
Pak kepala desa tidak tahu bahwa 3 pencuri di rumah itu adalah teman lamanya!
```

After case folding, non-alphanumeric removal, stop words removal, and stemming:

```
main peran 12 domba pentas
alayisme tetap jakarta
tidur kemarin
pin bb b12a3fc
bbm naik warga sedih
blokir website
```

```
karakter ga jelas non ascii appeared
pak kepala desa tahu 3 curi rumah teman
```

1.3 Issues and Discussion

When you encounter issues in using Kemangi, please report it to Kemangi's repository. You need GitHub account to do that. Don't worry if you don't have it, just create one. GitHub is a community full of generous programmers.

1.4 Support Kemangi

Visit http://bit.ly/kemangi-feedback for feedback and comments. Kemangi developers would love to hear your story. If you like Kemangi, don't forget to star it at upper right Kemangi's repository (need GitHub login).

CHAPTER 2

Installation

Kemangi runs on Java, which runs on any Operating Systems. You need to have at least Java version 7 installed in your system.

Follow these steps to get Kemangi into action:

1. Download Kemangi.
2. Download and install Java Runtime Environment.
3. Run Kemangi:

- For Windows user, right click on downloaded Kemangi application, and select "open with Java".
- For Linux user, you can invoke Kemangi via command line:

```
java -jar path/to/kemangi-VERSION.jar
```

CHAPTER 3

Basic Operation

3.1 Overview

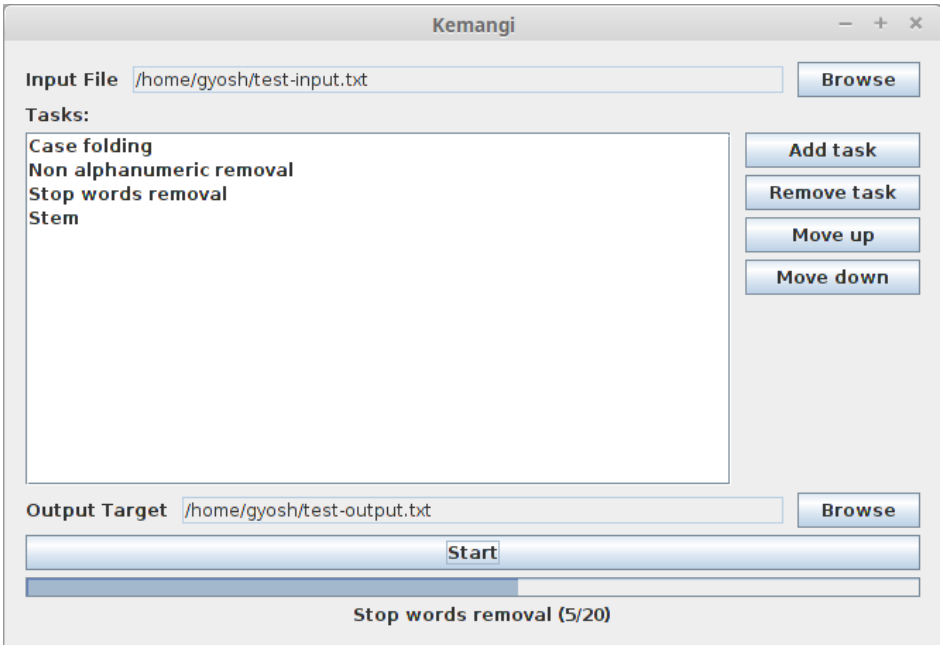
Kemangi takes a plain text file, and process each line of the text as your command.

Plain Text File A text file that can be opened using basic text editor in your system (Windows: notepad, Linux: gedit, vi, etc). If it shows up cleanly, then it is a plain text file.

Typical plain text file has .txt extension.

To do text preprocessing, you need to supply input file, list of tasks, and output target.

Here is a screenshot of a running Kemangi process to give a better understanding:



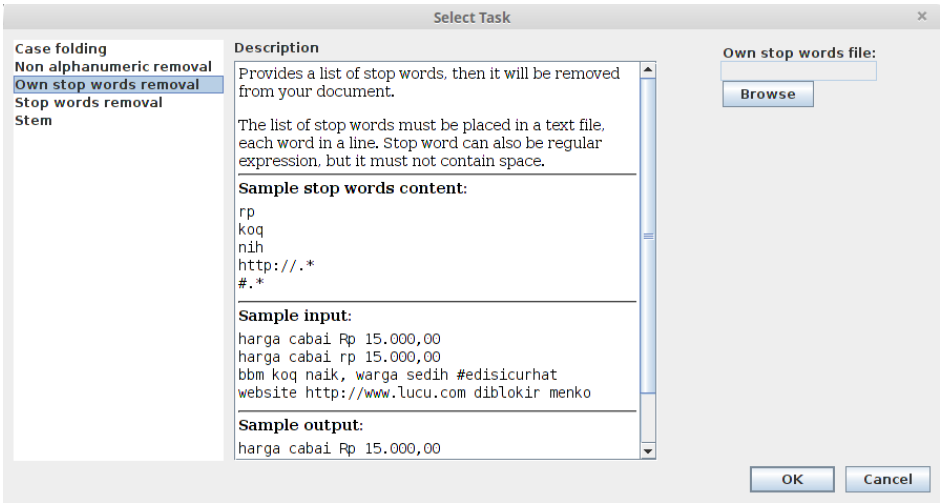
3.2 Input File

Simply by clicking "Browse" button. Remember that the file must be a plain text. Kemangi can't process more complex extensions like .doc or .odt.

3.3 List of Tasks

You can add several tasks, and Kemangi will run the task in the given order.

To add a task, click “Add task”. A window will appear and you can choose what kind of task to be added.



In general, there are three columns:

- Left column shows a list of available task, click one to choose.
- Middle column shows summary of chosen task.
- Right column shows additional inputs required. You need to supply these inputs if any is required.

To remove a task, click on that task in the list, then click “Remove task”.

To move a task’s order, click on that task in the list, then click “Move up” or “Move down”.

Some tasks may need additional inputs. For example, own stop words removal demands text file containing list of stop words provided by you.

Some tasks may also includes another tasks. For example, stop words removal already includes case folding.

3.4 Output Target

Simply by clicking “Browse” button. Then proceed as if you are going to save a file as usual.

3.5 Start Processing

Click the “Start” button and text processing will start.

When unexpected error occurs (required internet, but your connection is lost), Kemangi will print the latest preprocessed text `intermediateResult.bak` in Kemangi’s directory. It is a plain text file, and you can open it with any basic text editor.

Kemangi also logs its activity in `kemangi.log`. Log is useful to analyze what is going on when error happens.

CHAPTER 4

List of Supported Task

Each task is like a filter pipe. It receives input from one end, filter it, and outputs to the other end which is possibly another pipe's entrance. With this pattern, several pipes can be arranged successively, creating a workflow that you can customize.

Here are kind of task that Kemangi can do:

4.1 Case Folding

4.1.1 Description

Converts all uppercase letter into lowercase.

4.1.2 Example

Sample input:

```
Saya bErmain Petak Umpet
PAK MAU LAPOR
```

Sample output:

```
saya bermain petak umpet
pak mau lapor
```

4.2 Non-Alphanumeric Removal

4.2.1 Description

Removes all non-alphanumeric characters.

Recall that alphanumerics are letters and numbers. This operation may split a token into multiple tokens, e.g. `stop!minyak` becomes `stop` and `minyak`.

4.2.2 Example

Sample input:

```
Saya tidur... kemarin
TOLONG DIBENAHI!!! kapan beresnya??!?
```

Sample output:

```
saya tidur kemarin
TOLONG DIBENAHI kapan beresnya
```

4.3 Own Stop Words Removal

4.3.1 Description

You need to provide a list of stop words, then it will be removed from your document.

The list of stop words must be placed in a text file, each word in a line. Stop word can also be regular expression (regex) pattern, but it must not contain space.

If you are not familiar with regex, find out more here.

Typical used regex pattern:

Type	Example	Regex pattern
URL	http://www.doge.com, https://www.wow.com	`https?://.*`
Hashtag	#wow, #curcol, #love	`#.*`
Mention	@will.gozali	`@.*`
Numbers	021, 56001, 2123123123	`[0-9]+`

4.3.2 Requirement

- A plain text file containing list of stop words or regex pattern to be removed from your document. One word per line.

4.3.3 Example

Sample stop words content:

```
rp
koq
nih
http://.*
#.*
```

Sample input:

```
harga cabai Rp 15.000,00
harga cabai rp 15.000,00
bbm koq naik, warga sedih #edisicurhat
telah blokir website http://www.lucu.com
```

Sample output:

```
harga cabai Rp 15.000,00
harga cabai 15.000,00
bbm naik, warga sedih
telah blokir website
```

4.4 Stop Words Removal

4.4.1 Description

Removes meaningless word for further processing like `di`, `saya`, or `dari`. Uses web service provided by Faculty of Computer Science, University of Indonesia.

Try it: http://fws.cs.ui.ac.id/StopwordRemoverSampleClient/index.jsp

This task includes case folding and remove non-alphanumeric characters.

Be warned, the word `tidak` (en: not) is also removed. Depending on what you are going to do next, removing this word may affect the result

4.4.2 Requirement

- Internet connection.

4.4.3 Example

Sample input:

```
Pak kepala desa tidak tahu bahwa 3 pencuri
di rumah itu adalah teman lamanya!
```

Sample output:

```
pak kepala desa tahu 3 pencuri
rumah teman
```

4.5 Stem

4.5.1 Description

For each word, change it to its root form. Uses web service provided by Faculty of Computer Science, University of Indonesia.

It is good to know that the inventor of Indonesian Language's stemming algorithm is the one behind this web service developer.

Try it: http://fws.cs.ui.ac.id/StemmerSampleClient/index.jsp

This task includes case folding.

4.5.2 Requirement

- Internet connection.

4.5.3 Example

Sample input:

```
Mempermainkan peranan 12 domba di pementasan
```

Sample output:

```
main peran 12 domba di pentas
```

CHAPTER 5

Developer Guide

This section describes Kemangi's architecture for developer who might be interested to contribute.

5.1 Tools

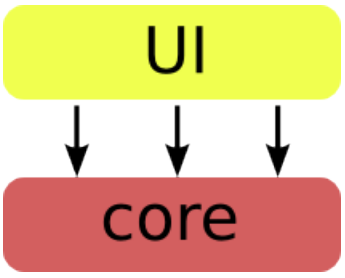
Kemangi uses Maven: Apache build manager for Java projects. By using Maven, dependencies, unit tests, and build options are configured under pom.xml.

5.2 Package Structure

Kemangi is mainly divided into two chunks: core and UI.

Core package basically does the work, while UI acts like an adaptor between user and core package. With this structure, core and UI may evolve independently.

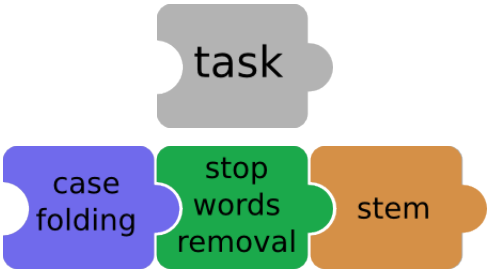
Another adaptor can also built on top of core modules, e.g. for Kemangi's support as command line tool.



5.3 Task Architecture

Each kind of task must implement interface `Task`. Task needs to have the capability to take a list of string tokens, do the job, and return a list of processed string tokens.

By using this architecture, tasks can be chained like a pipeline. User can easily customize what kind of task to be done and in what order.



5.4 Future Development

Check Kemangi's repository for issues and development plan. If you have any comment or suggestion, feel free to drop issues!