## WikiRules 2.0 - installation guide

## System Requirements

The development of the system was made using Intel ® Core ™ I7-2670QM CPU with 2.2Ghz (Quad Core) and 8GB of RAM memory and complete running on the English Wikipedia was made in about a week using the same machine as the Database server and application server. Therefore, any weaker machine might consume more time for the execution.

The software requirements are:

1. Installed MySQL server. (Development was made on version 5.5.27)
2. Java version 1.6.
3. JWPL (Java based Wikipedia Library) for creating and filling the Wikipedia database. Full instruction on how to get a full Wikipedia JWPL database from Wikipedia dumps can be found on <http://code.google.com/p/jwpl/wiki/DataMachine>
4. Python 2.7 (For EasyFirst English parser)

## How to build the rules database

1. Create the system scheme using the script "CreateDB.sql" supplied under the "DB\_Scripts" folder.
2. Fill in the Wikipedia Miner configuration file according to your requirements. The file is divided to modules in which every one of them responsible for specific number of parameters. The modules are self-explained. The most important module which you probably want to edit is "Extractors" which determine the extractors that will be used to fill the database. Other modules are the JWPL database configuration, the target Database configuration (which will be filled with rules) and "processing\_tools" which determine which processing tools will be used. Note, that the classpath should refer to the libraries of the configured tools - the current configuration for instance, requires lingpipe, stanford-postagger, stanford-ner, opennlp-tools, gate jars.

In addition, the parameters of the stopwords file path and the JARS folder path are at the top of the configuration file as ENTITY.

**\*Important note –** do not fill the database using both the lexicalIDM and syntacticIDM extractors. Using them both can result in wrong classifiers ranks.

1. The system uses the log4j framework as a logger mechanism. Make sure you have a log4j configuration file (log4j.properties) in the log4j directory. You can change the logger configuration as you wish)
2. For the English EasyFirst parser which run in a server-client manner run the server side on port 8081.
3. Run the system using

Java-Xmx<*Allocated Memory Size*> eu.excitementproject.eop.lexicalminer.wikipedia.MinerExecuter <*Configuration file path*>

\* We set *"Allocated Memory Size"* to 4000M but bigger value can reduce the running time.

\* The system has recovery mechanism which can recover from crash and skip the data which already been processed. In case of a crash all you have to do is to run the system again and mechanism will be used automatically.

1. During the execution you can view the log files which will be written the location defined in the log4j.properties file.
2. After a full success execution you have a database contains all the rules that extracted by the system. This database doesn't contain some indexes which important for retrieving the rules. To add those indexed run the script "CreateIndexes.sql" from the DB\_Scripts folder.
3. In this point you have a full database ready to use by the retrieval tool. If you choose to run the lexical or syntactic extractors and wish to use an offline classifier, please go the "Build the offline classifiers" section, otherwise, you can skip to "How to retrieve rules" section.

## Build the offline classifiers

1. The DB now contains all the rules and indexes, but the statistical data should be collected in order to run the classifiers. In order to gather this statistics run the "CollectStatictics.sql" from DB\_Scripts folder. (This script can take a while)
2. To run the offline classifiers you need to choose which of the supplied classifiers you want using the classifiers configuration file.
3. Run the classifiers ranks calculation process using

Java-Xmx<*Allocated Memory Size*> eu.excitementproject.eop.lexicalminer.definition.classifier.BuildClassifiers <*Configuration file path*>

An example of such configuration file is given at: /src/eu/excitementproject/eop/lexicalminer/definition/classifier/ BuildClassifiersConfig.xml

1. The database is now ready to retrieve rules using the offline classifiers.

## How to retrieve rules

1. Define the desired classifier and the other configurable parameters in the retrieval configuration file.

An example of such configuration file is given at: /src/eu/excitementproject/eop/lexicalminer/LexiclRulesRetrieval/ wikipediaLexicalResourceConfig.xml

1. Use the LexicalResource interface with an instance of WikipediaLexicalResource that get the retrieval configuration file path as a parameter.

An example of this usage is given ain the main method of eu.excitementproject.eop.lexicalminer.LexiclRulesRetrieval.WikipediaLexicalResource class.

**Created by: Dov Miron and Alon Halfon**

**Feb 2013**

**Modified by: Meni Adler**

**Sep 2013**