# Software Documentation

The aim of the software is to download necessary files from DBpedia data dump and static extraction of abbreviations, their meanings (including disambiguations, if there are multiple meanings), Wikipedia abstracts, categories, types and vocabulary extensions. Extracted data will be modelled using the Lemon Ontology, which will be queryable via SPARQL.
It is done for 119 languages.

The software consists of three files :

- a downloader script
- an extractor script
- a shell script

Here is the detailed description of all the scripts:

**1. Downloader Script** - This is a python2 script which downloads the necessary files from DBpedia data dump (http://downloads.dbpedia.org/3.9/) and decompresses them. The files are in bz2 zipped format which are unzipped from the script. It extracts the abbreviations ending with '.','!' or '?'.
It downloads and decompresses following files:

1. redirects - contains abbreviations and meanings
2. instance_types_heuristic - contains types
3. intance_types - contains types
4. interlanguage_links - contains vocabulary extensions
5. interlanguage_links_chapters - contains vocabulary extensions
6. labels - contains labels
7. article_categories - contains categories
8. disambiguations - contains disambiguations for the abbreviations

Libraries needed: urllib2 bz2, sys, re
Instruction to run the script: python downloder_extractor.py language_code

**2. Extractor script** - This is a python3 script which reads the redirects file downloaded by the downloader and extracts types, labels, categories and vocabulary extensions from the local endpoint via SPARQL queries, storing them to dictionaries. Then it writes the extracted data into a lemon file and a tsv file.
Libraries needed: sys, getopt, os, collections, SPARQLWrapper, JSON, re, urllib.parse
Instruction to run the script: python3 extractor_lemonMaker.py language_code

**3. Shell script** - It is a bash script which automates the complete process for all the 119 languages in the DBpedia data dump. It runs the downloader code, then it loads all the ttl files into local virtuoso graph set up in the system, runs the extractor code and finally deletes the graph from virtuoso and all the files downloaded.