Japanese glossary extractor

Japanese texts that are presented to students as a means of study require that the reader is familiar with most of (or all) the vocabulary contained in the text. An efficient way to present new words is by means of a glossary that contains new words, their meaning and grammatical properties. Glossaries should contain only vocabulary that the student has not yet learned, which is often not the case. Filtering the right items can be tedious, which is why I created a program that can automatically generate a vocabulary list from a given text, while taking the language level of the reader into account.

**Features:**

Only vocabulary equal to, or higher than the specified level will be shown.

Requires only one click to run. Very fool-proof.

If no level is specified, the script will check the level of the text and modify the input file’s title.

Currently uses 3 rating systems: JLPT, Genki and Minna no Nihongo.

High customizability:

Output options: Words, alternatives, level, PoS, reading, meaning

By default, alternatives, reading and meaning are exported.

The output file has a text box in which the source text can be pasted.

Texts can be filtered on customized references if the data files are provided.

Future update: new reference data for Intermediate Japanese.

Future update: extended data for Genki.

Future update: resizing the tables and fonts to increase the available space.

Future update: the output file combines the source text and glossary in a single .docx document.

**User input:**

A .txt file with the Japanese text

**Output:**

A .docx file with a filtered glossary. The displayed features can be customized.

**Data:**

Official reference files for JLPT, Genki and Minna no Nihongo.

Custom files should be in UTF-8 and can be formatted as desired. CSV and tables will work.

Data will be extracted from Jisho.org, so an internet connection is required.

**Processing**

The script starts by using the filename of the input file to generate a list with reference files belonging to that particular grading system. This is necessary because depending on the type of reference fine, the files may be named in reverse order, i.e. JLPT files are ranked N5 to N1 in increasing difficulty, whereas most reference files use chapter numbers to indicate difficulty. Most reference files are validated files that can be used for several purposes, so they should not be modified.

Once the dedicated vocabulary and kanji files are found, a dedicated tokenizer module tokenizes the text and filters out all grammatical function words, particles and punctuation. It generated a dictionary in which each word is tagged with PoS as well.

Next, the output from the tokenizer is reformatted and every word is reduced to its lemma. Placeholders for every word’s features are added to the dictionary, to avoid the misplacement of data later on, and to provide the tokenizer’s output as backup in case Jisho cannot find the word, i.e. proper names are already given final features at this stage, since Jisho may be unable to find any data for them.

Once the token dictionary is completed, a separate function looks up each word in the vocabulary files from easy to hard. If the file is not found, it splits the word into components and looks up every component in the kanji files, after which the highest found level is injected into its predefined slot in the dictionary.

After this stage, the program checks whether the input file was given a level to filter on. If this is not the case, the script uses the levelled dictionary to determine the level that has the highest amount of words. The suggested level is then added to the original filename. The script then ends, and the user can leave or modify the level in the input file.

If the script finds a level in the input file, all entries in the dictionary that are of a lower level are removed.

Finally, the lookup function iterates through this list and sends a query to the Jisho.org’s API. The received JSON file is analysed, and all data is filtered and accumulated in variables prior to the injection in the dictionary.

Finally, the output function generates a .docx file with appropriate title and headers, and a customized table with the final glossary.

**Example**

**Input:**

Txt file **“test file\_JLPT”** with content:

**“受賞決定で初めて知った。これから読む楽しみが増えました。”** (JLPT N5-N4-N3-N2-N1)

(I first learned of [the book] after it was awarded. My reading pleasure has increased ever since.)

**Output:**

**First run:** the script suggests a reading level: N5

**Second run**: the script generates a list based on the specified level:

|  |  |  |
| --- | --- | --- |
| Word | Reading | Meaning |
| 受賞 | じゅしょう | • winning (a prize) |
| 決定 | けってい | • decision, determination • Decision (European Union) |
| 初めて (始めて, 甫めて) | はじめて | • for the first time • only after ... is it ..., only when ... do you ... |
| 知る (識る) | しる | • to be aware of, to know, to be conscious of, to cognize, to cognise • to notice, to feel • to understand, to comprehend, to grasp • to remember, to be acquainted with (a procedure) • to experience, to go through, to learn • to be acquainted with (a person), to get to know • to concern |
| これから (此れから, 此から) | これから | • after this |
| 読む | よむ | • to read • to count • to guess, to predict, to read (someone's thoughts), to see (e.g. into someone's heart), to divine |
| 楽しみ (愉しみ) | たのしみ | • enjoyment, pleasure, diversion, amusement, hobby • anticipation, looking forward to |
| 増える (殖える) | ふえる | • to increase, to multiply |