Kantan Jisho
A Modern, Lightweight Japanese-English Dictionary

David Noetzel
April 30, 2015

Abstract

Even as more and more services move online, there remains a need for offline resources for language learners. While widespread, internet access is by no means always available, particularly when traveling; it is this very problem which originally spurred my interests in offline dictionaries. However, most other freely available offline Japanese-English dictionary programs have cluttered interfaces and are full of features that are rarely used or done better by other programs. For the user that wants only a dictionary, without flashcards, study notes or other superfluous features, I created Kantan Jisho, a modern Japanese-English dictionary free of feature creep, which pairs a simple, straightforward interface with fast, powerful searching. The program is designed to be cross-platform and uses a free software license to allow anyone to freely share, fork and modify Kantan Jisho as they please.

The difficulty in creating an efficient Japanese dictionary program comes from the fact that the language uses three different writing systems, all of which are used alongside one another. In addition to this, Japanese rendered in Latin characters, called romaji, is often used to type Japanese text on a computer. The primary challenge facing Japanese-English dictionaries is how to efficiently search a large corpus which holds so much variation and how to neatly display a large quantity of information, without the result looking cluttered.

To create a fully-featured dictionary program like Kantan Jisho, all of these challenges regarding input handling, data storage, proper handling of non-ASCII encoding schemes both current and antiquated, parsing of source material, efficient searching, keeping dictionary sources up-to-date and presentation of search results must be tackled.

1 Introduction

The desire to build a better dictionary arose when I was to study abroad in Japan in the summer of 2013. The study abroad program warned applicants that given the large proportion of elderly host parents, internet access was not guaranteed at home, meaning that there was a possibility I would be unable to use my favorite online dictionary. I was reminded of the need for offline resources for language learners. Though its availability grows with every passing day, internet access is by no means a given in every situation; having options that do not require an internet connection remain valuable. As mentioned previously, the current crop of Japanese dictionaries are outdated, bloated or available only for one platform; I believe that a better option can and should exist; for that reason, I created Kantan Jisho.

In this paper I will cover the following topics:

- Section 2 discusses the background of electronic Japanese-English dictionaries.
- Section 3 discusses related work in Japanese-English dictionaries.
- Section 4 discusses the features of Kantan Jisho and what role they play.
- Section 5 discusses the design challenges encountered during the development of Kantan Jisho and how they were solved.
- Section 6 presents opportunities for further development in Kantan Jisho.
- Section 7 gives a conclusion.

2 Background

2.1 A Brief Introduction to the Japanese Writing System

To understand the unique challenges facing a Japanese-English dictionary, one needs to also have at least a surface understanding of the Japanese writing system. Unlike characters in Western alphabets, Japanese has three distinct writing systems: kanji, borrowed Chinese characters which are ideograms and number in the thousands; hiragana, a syllabary primarily used for conjugation, inflection, postpositions, and words for which no kanji exists; and katakana, a syllabary primarily used for loanwords, emphasis and onomatopoeia. These two syllabic
scripts share the same sounds and can be referred to generally as “kana.” In addition to these native writing systems, Japanese is sometimes rendered in Latin characters called romaji. There are multiple standards for Romanization, so to be effective, a dictionary should be able to parse all of them.

2.2 Dictionary Files

Almost all electronic Japanese-English programs and websites in the Western world are based off of just a few sources, most of which are from the Electronic Dictionary Research and Development Group (EDRDG); all files used in Kantan Jisho, unless otherwise noted, are from this organization. The most popular general dictionary containing a vast number of words, expressions and phrases is JMdict (Japanese-Multilingual Dictionary). The project started in 1991 with the release of a simple, free Japanese-English dictionary called EDICT. The file created by Jim Breen, formerly of Monash University, due to its free and open nature, was incredibly popular. In 1999, the project evolved into JMdict, an XML file whose features allowed for a much greater depth of information to be recorded, including support for multiple languages. compscidic, another file dedicated exclusively to computer terms, was rolled into JMdict, making a separate released unnecessary. Kantan Jisho uses the English-only JMdict.e release as its “General dictionary.”

Given that an average Japanese high school student will know more than 2,000 characters by the time he graduates, mastering kanji is often the most significant roadblock to a learner of Japanese. Adding on to the sheer number of characters is the fact that there are almost always at least two (and often more) ways to pronounce a particular kanji. It is understandable then that are occasions when intricate details about a specific kanji can be quite useful. This information includes what the character fundamentally means and how it is pronounced in a Chinese-derived context (on’yomi) and in a native Japanese-derived context (kun’yomi). Kantan Jisho uses kanjidic2, another XML-based product of the EDRDG, as its “Kanji dictionary.”

Japanese proper nouns are also tricky. Kanji can have obscure readings that are only found in the names of people and places, and plenty of these names can be ambiguous or confusing, even to a native speaker. Foreign names are often no better, with a single proper noun having many common Japanese renderings; a search for “Vladivostok” revealed six different pronunciations. Knowing how one could interpret these names can be critical to a learner, so Kantan Jisho includes JMnedict, another EDRDG offering, as its “Names and places” dictionary.

For specific science and engineering terms, Kantan Jisho uses engscidic. This is the only dictionary that is not updated on a regular basis. Compiled from a Japanese source, cleaned up and last updated in 2001, this dictionary, though static, provides a small but useful resource for technical and obscure words. In Kantan Jisho, it is the “Engineering” dictionary.

Tying these sources together is the Tanaka Corpus. One Professor Tanaka assigned his students to collect pairs of English and Japanese sentences. From this humble, error-filled corpus has come a set of hundreds of thousands of useful example sentences that have been thoroughly scrutinized and heavily corrected; the corpus is actively maintained by the Tatoeba Project. A learner of Japanese needs to have access to an ample corpus of the language to truly understand how words are used in the context of a sentence. The “Sentences” option in Kantan Jisho uses the Tanaka Corpus as its source.

3 Related Work

3.1 zkanji

zkanji is the program I was first drawn to when searching for offline dictionaries. Using JMdict and kanjidic2, it supported general word searching and kanji searching, but I found the interface to be cluttered and confusing, as the following screenshot demonstrates.

![zkanji's main interface](image)

Considering that the tiny dictionary in the lower right-hand corner is all I care about, zkanji was not for me.

3.2 Tagaini Jisho

Tagaini Jisho is the only other modern program I found, but I found the interface to also be unwieldy. In addition, its role as a “study tool” means that it contains a number of features that I found unnecessary and distracting.
4 Features

The design of Kantan Jisho is conceptually very simple. The program takes the user’s text input and search options, converts romaji input to hiragana if necessary, generates a database query, queries a database and displays the results of that query in a tabbed interface. There were however, many subtle issues that made this process much more complicated than it first appears. In this section, I describe some of the goals of Katan Jisho and how they were achieved.

4.1 Cross-platform Framework

Kantan Jisho is built on version 5.4 of the Qt framework using C++. When thinking about what I wanted to build, the Qt framework was an obvious choice. Qt allows for easy cross-platform development, with a focus on GUIs that will have a native look and feel on every platform. The framework also includes a huge range of tools that included everything that I needed to make Kantan Jisho work on the back-end: seamless support for Unicode strings, an XML parser and a means to create and interact with SQLite3 databases. Qt also software authors to use the GPL, a feature which I strongly support.

4.2 Wildcard searching

Learning kanji is often the most challenging hurdle for speakers of Japanese, native and non-native alike. Looking up an unknown kanji can be time-consuming, but there are situations in which the reader recognizes some of the characters in a word. Wildcard searches can use this partial knowledge to find a match. Just as in an SQL GLOB statement, ? represents a single character and * represents any number of characters. This allows a user that remembers only part of a word to search the bit he remembers and use the results from a wildcard search to identify an unknown word in less time than it would take to look up an unknown character.

4.3 Simple Interface

To avoid the cluttered interface seen in zkanji and Tagaini Jisho, I resolved to keep the user interface simple and to specifically avoid feature creep. Kantan Jisho means “simple dictionary” and this idea is at the core of the program’s design. Taking cues from the (formerly) expertly designed website Denshi Jisho, I wanted to have an interface that anyone could understand at a glance and that prioritized the search results. To that end, the interface is free of panels, dialog boxes and side-bars. A screenshot demonstrating the user interface of Kantan Jisho can be seen below.
4.4 Ability to Handle Multiple Inputs

As described earlier, the three writing systems of the Japanese language are a particular nuisance to a software developer trying to create a program that involves searching. Some words have kanji associated with them that are used less and less these days in favor of writing the word purely in kana (for example, “まっとう” (mattou; perfect)). Some words, particularly onomatopoeia, are written with either hiragana or katakana interchangeably (for example, “きらきら” (kirakira; glisten)); some words mix both hiragana and katakana in the same word (for example, “サボる” (saboru; to be truant)). This can particularly become a problem if the user is entering romaji: after all, a user entering “haiku” may be trying to find the characters associated with the traditional Japanese poem (俳句) or he may be trying to confirm that his search term is indeed the loanword for the English word “hike” (it is). This very type of scenario is why Kantan Jisho first converts all romaji input to hiragana and also performs kana searches on a character-by-character basis, ensuring that no matter what input is given, it can do its best to find the desired term with a minimum of effort on the user’s part.

4.5 Unobtrusive Automatic Updating

Kantan Jisho features automatic updating on a weekly basis. While most of the dictionary files are updated on a near-daily basis, the day-to-day changes are so small that updating to each new version is not necessary. However, the dynamic nature of language makes some sort of update mechanism necessary, as does the crowd-sourced, continually improving nature of the data in the dictionary files. If a week is too short for the user, he can force Kantan Jisho to check for dictionary updates at any time via the menu bar. Due to the server often being rather slow and due to the size of the files being processed, having the program be unusable during the update process would lead to an awful user experience. Thanks to the modular nature of the dictionaries, however, it was possible to disable only the dictionary file being updated, allowing the user to freely search the other dictionaries while the update progresses. In fact, it is likely that the only downtime a user will ever experience while using Kantan Jisho is 30-60 seconds upon starting the program for the first time; Kantan Jisho purposefully does not bundle any dictionaries and so the dictionary files all need to be downloaded upon the first run.

4.6 Search Filters

Given that JMdict stretches into the hundreds of thousands of entries, it is important that the user be given more control over what results are returned. While the possibility for more exotic filters exist, the two I decided to implement with my limited time were exact matching and hiding katakana-only words from the results. The default search for Japanese terms in Kantan Jisho is greedy; it merely finds words that start with the given term. This can be quite useful when one searches, for example, “大学” (daigaku, university) and also wishes to see related terms like “大学院” (daigakusei, university student), “大学” (daigakuin, graduate school), etc. However, particularly for common stems or single character words, turning this feature off is vital to not being overwhelmed by results. Take, for example, a search for the kana “は”; a common word with only this one syllable is “絵” (e, picture). With the exact search on, there are 14 results; with it off, there are 3,869. A similar justification can be made for hiding katakana-only words. Many of the words composed entirely of katakana are loanwords, in particular, English loanwords. To an English speaker searching for “はい” (hai, yes), the entries for “ハイ” (hai, high), “ハイアラーキー” (haiara-ki-, hierarchy), “ハイウエー” (haiue-, highway) and so on are often worthless.

4.7 Search History

As alluded to earlier, a search history can prove invaluable when trying to recall words. In Kantan Jisho, the search history is displayed very plainly in reverse chronological order, in a tab with the timestamp, the Japanese term in the English term. This provides the user with enough information to see what he was searching and when. Of course, for the sake of privacy, the search history can be cleared at any time.

5 Challenges

While there were some significant and time-consuming tasks in the development of Kantan Jisho, the bulk of the challenges can be more accurately described as a mountain of quirks and inconsistencies. In this section, I will present a sample of these problems.
5.1 Converting Romaji to Hiragana

Given that there are three different systems for Romanizing Japanese in wide use (with a few others filling particular niches), converting romaji to hiragana requires one to pick sides to a certain extent. Due to its intuitive, phonetically-based transcriptions, Hepburn romanization is the most widely used system, especially in the West; I decided to support it in full and add whatever features of other systems that made sense. Another factor in navigating the challenges of converting romaji is the letter “N.” In Japanese, it can be either a nasal N (終: [nj], as in finger) or a “regular” n (e.g. 聞: [nu], close to new). A nasal N can be indicated implicitly by an N followed by another non-N consonant or represented explicitly by two Ns or an N followed by an apostrophe. To solve this, I once again made the parser greedy. That is, if the user wanted to type “おんヨミ” (from onyomi; Chinese-derived reading of a kanji) in romaji, he would need to type “on’yomi” or “onnyomi.” “Onyomi” would result in “おにょみ,” (pronounced “o-nyo-mi”) which is a very different set of sounds to a native speaker.

5.2 Different Data Sources

Due to Kantan Jisho’s reliance on multiple sources, many different methods of handling input data had to be written. While JMdict and JMnedict shared a similar but not identical structure, and kanjidic2 and the two aforementioned dictionaries were XML files, there is no common format between each data source. There also existed quirks in each data source. JMdict contained entries which had readings bound to one specific set of kanji, which can sometimes be seen in a common word which has two readings, but also two different sets of kanji due to a complex, archaic character being simplified. The Tanaka Corpus contains a surprising number of exact duplicates that need to be weeded out before the database is populated, lest the primary key constraints be violated. The engineering dictionary contains entries where two fields in the same row would have the same content. All these small idiosyncrasies needed to be accounted for and thoroughly vetted to ensure that the data was accurate and sensible to a human user. Once this data was parsed and vetted, it was inserted into flat tables, one entry per row. While I acknowledge that this is not the most efficient method of storage, it was a compromise; sacrificing a few milliseconds of search time and some disk space was, in this timeframe, worth the increased simplicity of database queries. In the future, the database schema could be improved by using more tables linked together, with less redundant data.

5.3 Search Performance

In lieu of the choices described previously, achieving top-notch search performance could get rather challenging. Of particular difficulty was enabling a kana search that could search for mixed hiragana and katakana words. My original design implemented the REGEXP function in SQLite. SQLite does not have native support for this function, but it can be supplied by the user. So long as the program calls sqlite3_create_function() and provides a pointer to a function implementing the regular expression evaluation, the SQLite will be able to evaluate regular expressions like any other SQL-based DBMS. However, converting a C-style char* string into a QString and then using QRegularExpression (the newer and faster regex module in Qt that replaced QRegExp) to evaluate the regex was slow; a few of my tests indicated that it could be up to a full order of magnitude slower than the current search. Despite spending a great deal of time familiarizing myself with the SQLite3 API during this process, the code associated with generating queries using REGEXP had to be thrown out, setting me back significantly.

In place of the REGEXP operator, I opted to generate a series of nested queries. For example, if one wanted to search the kana “サボ” (saboru, to be truant), one could construct a much more efficient query like so (select * is used in this example for readability):

```sql
select * from
(select * from
(select * from words
where kana glob 'サボ*' or kana glob 'サボ**')
where kana glob 'サボ*' or kana glob 'サボ**')
where kana glob 'サボ*' or kana glob 'サボ**'
```

Using this method was a much-needed improvement. Incidentally, this also led to a curious discovery. In testing, I found that one could progressively build up a query for any Japanese term (even one consisting of kanji, where there is no alternate character to search) and the result would be slightly faster. Not knowing what is going on with the internals of SQLite, I can only guess why this is the case; given much of computing’s history with poor support for non-ASCII text encoding, it would not surprise me in the least if this difference in performance was due to poor handling of Unicode characters.

6 Further development

While much has been accomplished with Kantan Jisho, there are of course more opportunities to improve and expand upon the software. I do plan to continue development on Kantan Jisho past graduation, so others can use and contribute to the codebase.

6.1 More Search Filters

In the effort to be complete, JMdict has thousands of words that are rarely ever used in modern Japanese: archaisms, jargon, hip English loanwords that are rarely used outside of pop music and advertising campaigns,
etc. The comprehensiveness can certainly be useful, but many times it makes the dictionary cumbersome. The previously mentioned Denshi Jisho has a feature that allows users to filter out uncommon words; this has proved to be my favorite feature of the site, but information on how this filter was achieved seems hard to come by. I have found no files published by the EDRDG nor any features in any of the dictionaries that would indicate some sort of frequency analysis or designation of common usage, nor does the site indicate how it designates a word "common." If I find such data, it will assuredly be the first new feature to be added to Kantan Jisho.

6.2 Improved Presentation of Search Results

Currently, Kantan Jisho makes poor use of horizontal space. Given that the majority of computer monitors these days have an aspect ratio of 16:9, the rather thin profile of Kantan Jisho is certainly a weak point. With a little more time, the search results could certainly be displayed in a prettier fashion that better suits the state of desktop applications in 2015. That said, there is a fine line that one must tread between the well-designed minimalism Kantan Jisho champions and a results tab that puts form above function.

6.3 Optimization

There are many optimizations that can still be made in Kantan Jisho. The primary source of noticeable improvements would come from the potential table rearrangements discussed in section 5.3. Such a move would go a long way to speeding up database queries. Most other optimizations would be far more minor. For example, while Kantan Jisho typically uses very little memory, its memory usage skyrockets during dictionary updates, since currently downloaded files are saved entirely to RAM before being written to disk. On top of this, the processing of files is also currently done entirely in RAM. While not a problem for a simpler file like the Tanaka Corpus, this could be a big problem on lower-end machines when processing JMnedict, the largest one of the five.

6.4 Non-English Languages

I had originally planned for this to be supported in Kantan Jisho, but after looking at the actual files, I decided against it. Languages other than English are supported only in JMDict and kanjidic2. In these dictionaries, the number of entries with non-English definitions is so paltry as to not be worth supporting. If say, a German were looking to use Kantan Jisho to search JMDict in his mother tongue, he would be sorely disappointed. Even if the option did exist, having such a low percentage of words with definitions in anything other than English would simply provide a bad user experience. I believe strongly that people should be able to use software in their native languages, but unfortunately, this is an issue out of my control. If the sources ever improve to the point where adding non-English languages makes sense, I will very gladly add that feature.

7 Conclusion

Kantan Jisho provides a great new alternative to the current crop of Japanese-English dictionaries. By focusing on the basic search experience and not getting bogged down in extraneous features or a busy user interface, it is in a great position to expand and develop further. While there are still ways in which Kantan Jisho can be improved, this semester has already born fruit and has also laid a very solid foundation for future work.