Japanese Helper

Term Paper for
Industrial Strength Multilingual Language Analysis 2017-18

Verena Blaschke and Xiang Ji

March 26, 2018

Contents

1 Introduction

Japanese is studied by many non-native speakers. One of the biggest challenges for beginners are kanji–Chinese characters that are used to spell many content words–because their pronunciation is difficult to predict since it depends strongly on context. Therefore, we present this tool that aims to help Japanese learners by tokenizing any given input, and providing pronunciation information, translations and morphological information for each token, all visible at the same time.

This paper is structured as follows: In section 2, we present existing web tools with similar functionalities. Then, in section 3, we list the project dependencies. In section 4, we describe how we preprocess the Wiktionary dump. This part consists of several major subsections: First, section 4.1 provides an overview over the Wiktionary dump we work with and explains the basic preprocessing. Second, section 4.2 gives an introduction to inflection in Japanese to then explain how we generate inflected word forms. Third, section 4.3 explains how we link the preprocessing step to the server startup phase. In section 5, we detail how we process the user input, and in section 6, we describe the user interface. Section 7 gives an overview of the division of tasks. Finally, section 8 provides a discussion of the tool.

2 Comparable Tools

Other tools with similar functionalities exist:

- *Jisho*[1] is an online Japanese-English dictionary that can also process phrases and sentences. Given a phrase or sentence, it tokenizes it and presents it with along with pronunciation information for the kanji and some simple part-of-speech (POS) information (verbs, nouns (and names and interjections), adjectives (and adverbs), and particles are each displayed with a different font colour). *Jisho* also displays dictionary information on the first token in the user input, and the user can click on the other tokens to see their dictionary entries (if available). This dictionary information also includes a difficulty rating, and, if the word is a verb or verbal adjective, the user can open an inflection table pop-up. If the user selects an inflected token, the website presents the dictionary entries for the lemma form, as well as information on the particular inflection.
- The Japanese-to-English version of *Google Translate*[2] mainly provides automatic translations of the user input, but it also gives some tokenization, and it provides pronunciation information by displaying a Latin transliteration of the input. Additionally, it gives more information about some of the content words in the input text (synonyms, POS information). To get this additional information, the user has to manually select different words in the input window.
- *WaDoku*[3] is an online Japanese-German dictionary that is similar to our tool in that it presents the tokenized user input in a table alongside possible translations and pronunciation information. It also gives the pitch accent and offers the decomposition of compound nouns and inflected words. The results are presented in a mostly vertical table that contains all matches for each token. The user can select a match to be redirected to that word's entry in the dictionary. If it is a verb or verbal adjective, the user can open a table enumerating the word's inflection bases.

Our tool is different from *Jisho* and *Google Translate* in that it presents detailed information for all tokens in the user input at once. The learner can just look for the piece(s) of information that they want to know, without needing to select different tokens. It is also differs from

[1] Available at `http://jisho.org`
[2] Available at `https://translate.google.com/?tl=en&sl=ja`
[3] Available at `https://www.wadoku.de`.

WaDoku by giving English translations instead of German ones and displaying results in a primarily horizontally-motivated fashion, so as to prevent scrolling as far as possible. Unlike *Google Translate* and *WaDoku*, we also provide full inflection tables.

3 Project Dependencies

The project dependencies are managed by Maven. We used the following Maven packages:

- Google Web Toolkit and gwt-maven-plugin (Version 2.8.1)
- Kuromoji (Version 0.9.0), an open-source Japanese morphological analyzer. It is used for tokenizing the user input. (see Atilika (2018))
- Guava (Version 24.0)[4], Google core libraries for Java. Its `Multimap` class is used for storing the tokens created form the Wiktionary dump.
- Kanatools (Version 1.3.0)[5], a tool for working with kana. It is used for the conversion between katakana and hiragana.
- OpenCSV (Version 4.0)[6], a Java CSV library. It is used for reading the list of 常用漢字 (*jouyou kanji*), which is used as information source for kanji difficulty ratings.

4 Preprocessing the Wiktionary dump

4.1 Reading in the Wiktionary information

The Wiktionary dump[7] that we work with was compiled by Johannes Dellert and is based on entries about Japanese words in the English Wiktionary (see Wikimedia Foundation (2018)). It contains a little over 70.000 tokens. Every token entry contains information about its spelling (usually in kanji and/or kana[8]), its pronunciation (usually in hiragana), its POS tag, and a list of its English translations, as shown in Figure 1. The POS tags for many verbs and adjectives are followed by the name of the word's inflection paradigm. Additionally, they often contain transitivity information about verb entries.

```
計算言語学  けいさん げんごがく  N  1) computational linguistics
計算高い  けいさんだかい  A[i]  1) calculating
計里  けり  N  1) [spelling variant of 鳧]
計量  けいりょう  N  1) measurement; computation 2) metric [mathematics, physics]
計量する  けいりょうする  V[suru]  1) measure, gauge, weigh
```

Figure 1: An excerpt from the Wiktionary dump.

We read the Wiktionary file and store each entry as an instance of `Token`. This class contains fields for the form, pronunciation, POS tag, inflection class and the list of translations. It is extended by two subclasses: `InflectableToken` and `InflectedToken`. For tokens with inflection paradigms, we additionally generate inflected forms. This is discussed in more detail in section 4.2.4 below.

When creating a `Token`, we clean its list of translations as well as the POS tag:

[4] `https://github.com/google/guava`
[5] `http://mariten.github.io/kanatools-java/en/kana-converter/`
[6] `http://opencsv.sourceforge.net/`
[7] It is saved as `src/main/webapp/WEB-INF/dictionary/ja-wiki-preprocessed-20180309.tsv`.
[8] The two syllabaries hiragana and katakana (collectively called *kana*) are commonly used writing systems for writing Japanese, besides kanji.

We convert the string containing the enumeration of translations into a list of strings. If such a list begins with an entry that is marked as *archaic* or *obsolete*, we re-order the list so it begins with a more commonly used meaning. Additionally, we un-escape HTML characters. We further remove remains of the Wiktionary markup from the strings by removing empty parentheses, unnecessary curly braces, repeated whitespace, and by transforming the markup of sub-enumerations.

The POS information is compressed in the Wiktionary dump. To display more informative information to the user, we translate these into more verbose strings that we store in an additional field of `Token`. For most POS tags, this is very straightforward, e.g. we represent *ADV* as *adverb* and *N* as *noun*. If a POS tag is followed by a inflection paradigm, we also add that information, e.g. we represent *A[na]* as *adjective (na infl.)*. The POS tags for many verbs also include additional information on transitivity and inflection classes, as shown in Table 1. For instance, the verbose version of *VT1[go-mu]* is *transitive godan verb (go-mu infl.)*.

part of speech	**transitivity (optional)**		**inflectional group (optional)**		**other (optional)**	
`V`	`I`	intransitive	`1`, `5`, `godan`	godan	`form`	the entry is
	`T`	(di-)transitive	`2`	ichidan		already inflected
	`B`	ambitransitive	`3`	irregular		
			`ni`	nidan		
			`yo`	yodan		

Table 1: Elements of part-of-speech tags for verbs

We save the tokens in a `MultiMap` from word forms to `Token` objects. This format is useful later when we attempt to create the best possible match between the words from the user input and the entries from the Wiktionary dump (more on this in section 5). Since some of the forms and pronunciations of Wiktionary entries (as well as some of the suffixes from the inflection table templates) include whitespace, periods or dashes, we remove these from the token forms and pronunciations before adding the token to the collection. If the form or pronunciation of a *na* adjective ends with "(な)" *na*, we remove this ending since many *na* adjective entries in the Wiktionary dump do not have it, and because it is not needed for generating the inflections.

4.1.1 Kanji difficulty information

One thing that can be helpful to learners is information on the difficulty of the words they are reading. The Japanese Ministry of Education maintains a set of 2.136 常用漢字 *jouyou kanji* (literally "Chinese characters of general use") (Iwasaki, 2013, p. 24), which are to be taught in compulsory education curricula. In addition, official government documents are required to only use those kanji characters. We were able to retrieve the list of 常用漢字 from Wikipedia[9], with the school grade at which each kanji is supposed to be taught attached. When a token is created from the Wiktionary dump, additional information on the difficulties of the characters that constitute the token is added. Ratings for different characters are connected with a dash (-), and the ratings for non 常用漢字-characters (including kana) is represented with an asterisk (*).

[9] `https://en.wikipedia.org/wiki/List_of_jōyō_kanji`

4.2 Generating inflected forms

4.2.1 Inflection in Japanese

Japanese is an agglutinative language, as can be seen when analysing verb and adjective inflection. There are two types of adjectives in Japanese: verbal adjectives, which inflect, and nominal adjectives, which do not inflect and are instead followed by the inflected copula (Iwasaki, 2013, pp. 61–62). Verbs and verbal adjectives can take suffixes that indicate tense, aspect, voice, modality, polarity and register (Iwasaki, 2013, pp. 15–16, 89, 126). There is no agreement between heads and dependents however (Shibatani et al., 2017, p. 20).

Arguments and adjuncts are followed by case-marking particles that are analysed as separate words (instead of inflectional suffixes) (Iwasaki, 2013, p. 104–105, 117–120). This is why we only generate inflected forms of verbs and adjectives, but not of nouns.

Japanese verb and adjective inflection works by adding one or more suffixes to a word's stem (Iwasaki, 2013, pp. 78–79). Different stems exists; they vary with respect to the suffix that follows the root (Iwasaki, 2013, pp. 78–79). Each inflection base corresponds to a different stem (although there is syncretism between some of the stems). Table 2 shows the inflection bases that exist in modern Japanese. Each additional suffix that is directly added to the stem requires a specific base (McClain, 1981, pp. 5–6, 39).

no.	**Japanese term**	**term used in the Wiktionary inflection table templates**
1	未然形 *mizenkei*	imperfective
2	連用形 *renyoukei*	continuative
3	終止形 *shuushikei*	terminal
	連体形 *rentaikei*	attributive
4	仮定形 *kateikei*	hypothetical
5	命令形 *meireikei*	imperative

Table 2: Inflection bases for verbs and verbal adjectives (McClain, 1981, pp. 5–6), (Iwasaki, 2013, pp. 78–79)

There are three groups of verbs in modern Japanese: 五段 *godan* ("five-row") verbs, 一段 *ichidan* ("one-row") verbs[10] and irregular verbs (Hamano and Tsujioka, 2011, p. 53). They are, respectively, also known as *type I*, *II* and *III* verbs (Hamano and Tsujioka, 2011, p. 53). Wiktionary uses a combination of these naming conventions to refer to the different groups (see Table 1 and the list of inflection table templates in section 4.2.2).

Godan and ichidan verbs differ in how much of their lemma is considered to be their root, and in the suffixes that are appended for creating the inflection base stems. Godan verbs end with *-u* and their root is analyzed as the entire lemma excluding the final vowel[11] (Iwasaki, 2013, p. 78). Ichidan verbs end with *-eru* or *-iru* (Iwasaki, 2013, p. 78). Their root is the lemma without the final る *ru* (Iwasaki, 2013, p. 78). The third group encompasses 来る *kuru* ("to come/go (towards the speaker)") and する *suru* ("to do") (including verbs that were derived from other words by adding -する) (McClain, 1981, pp. 10–12).

[10] Respectively also known as consonant (root/stem) verbs and vowel (root/stem) verbs (Iwasaki, 2013, p. 78), (McClain, 1981, pp.6–8).

[11] Note that if the final syllable of such a verb has the structure *CV*, extracting the root means splitting the final syllable. This is not possible when using hiragana. Therefore, we (and the Wiktionary inflection table templates) consider the *entire* final syllable the stem-forming suffix when generating inflection bases and inflected forms.

A few verbs in the Wiktionary dump are marked as 二段 *nidan* ("two-row") or 四段 *yodan* ("four-row") verbs in their POS tags (see Table 1), but they are not associated with any corresponding inflection table templates[12]. These verb groups existed in Classical Japanese, but the verbs that used to belong into these categories evolved to be Modern Japanese ichidan and godan verbs, respectively (Komai, 1979, pp. 19, 28, 33). Most of the nidan and yodan verbs in the Wiktionary dump either lack any translation or only have translations that are marked as *obsolete*.

The generation of inflected word forms is tested in `InflectionTest`.

4.2.2 Inflection table templates

The inflection table templates that are used on Wiktionary are used for generating the different bases as well as complete inflections. This includes all of the inflection bases mentioned in Table 2, as well as ten inflections for verbs and thirteen for adjectives (verbal and nominal)[13]. Some of the forms are missing in some of the templates, whereas many templates include multiple alternative suffixes for specific inflections. We store the names of the inflection bases and forms in an enumeration called `Inflection`.

Most verbs and adjectives in the Wiktionary dump are associated with inflection paradigms for which inflection table templates from Wiktionary are available. We saved eighteen such templates in the resources folder[14]:

- the godan verb inflection table templates *go-bu*, *go-gu*, *go-ku*, *go-mu*, *go-nu*, *go-ou*, *go-ru*, *go-su*, *go-tsu*, *go-u*,
- the ichidan verb inflection paradigm *ichi*,
- the additional verb inflection table templates *honorific*, *suru*, *suru-i-ku*, *suru-tsu*, *zuru*,
- the inflection group for verbal adjectives, *i*,
- and that for nominal adjectives, *na*.

However, similar templates do not exist for some irregularly conjugated verbs and two rare types of nominal adjectives. We therefore manually created inflection table templates for the inflection paradigms that some Wiktionary entries refer to, but for which no templates exist on Wiktionary: the template *kuru* for the verb 来る *kuru* ("to come (towards the speaker)"), and templates for the nominal adjective groups *nari* and *tari*. Furthermore, we replace the inflection paradigms for certain entries and instead use templates for them that we created: instead of *verbconj*, we use *aru*, *beshi*, *da*, *dekiru*, *iku*, *kureru* and *ya*, and we replace *verbconj-auto* with *suru-indep*. We also change the name of the *ichidan* class to *ichi* during this step, but we did not need to create a new template for this group. Table 3 gives more details. The templates for *nari* and *tari* are both based on Martin (1975, p. 749) and Stalph et al. (2009, p. 20); the others are based on the conjugation tables belonging to the entries' Wiktionary pages[15].

[12] However, two yodan verbs are assigned godan inflection table templates.

[13] For verbs, these inflections are *passive*, *causative*, *potential*, *volitional*, *negative*, *negative continuative*, *formal*, *perfective*, *conjunctive*, and *hypothetical conditional*. The copula template additionally has *formal perfective*, *formal volitional*, and *formal conjuncitve* forms. For adjectives, the inflections are *informal negative*, *informal past*, *informal negative past*, *formal*, *formal negative*, *formal past*, *formal negative past*, *conjunctive*, *conditional*, *provisional*, *volitional*, *adverbial*, and *degree*. Where there is no mention of tense, aspect, register, or polarity in these inflection names, but such information can apply, they are *non-past*, *non-perfective*, *plain*, and *affirmative*, respectively.

[14] Each these templates is taken from `https://en.wiktionary.org/w/index.php?title=Template:ja-`<*inflection paradigm*>`&action=edit` and saved as `src/main/webapp/WEB-INF/ja-`<*inflection paradigm*>`.txt`.

[15] They can be found at `en.wiktionary.org/wiki/`<*lemma*>`#Conjugation`. For *aru*, *kureru* and *suru*, we used the Wiktionary pages describing the hiragana entries.

original class	new class	entries
verbconj	aru	有る, ある *aru* ("to be, exist", used for inanimate objects)
	beshi	べし *beshi* ("must, shall")
	da	だ *da* (copula)
	dekiru	出来る *dekiru* ("to be able to do")
	iku	行く *iku* ("to go, come")
	kureru	呉れる, くれる *kureru* ("to give")
	ya	や *ya* (copula (Kansai dialect))
verbconj-auto	suru-indep	為る, する *suru* ("to do")
ichidan	ichi (already a Wiktionary inflection paradigm)	泣き崩れる *nakikuzureru* ("to break down crying")

Table 3: Changes to the inflection paradigm structure. Entries that were originally assigned the inflection class *verbconj* or *verbconj-auto* have irregular inflection forms.

The inflection templates mainly consist of key-value pairs, where the key is the name of the base or inflection and the value consists of the suffix[16] that is added to the root. (Exceptions to this are explained in section 4.2.4.) For each inflection table template, we create a map from `Inflection`s to suffixes. Then, we fill that map by converting the key-value assignments in the template into `Inflection-String` pairs.

Because most of the verb inflection templates lack the *formal negative*, *formal perfective*, and *formal negative perfective* inflections despite their being common, we add them to the inflection maps in question. Generally, the formal form of a verb is its 連用形 *renyoukei* base followed by a form of the helper verb ます *masu*, where ます can be inflected for the desired grammatical categories. To add the *formal negative*, *formal perfective*, and *formal negative perfective* forms to an inflection map, we simply take the suffixes belonging to the *formal* form and, respectively, replace ます with its inflected forms ません *masen*, ました *mashita*, and ませんでした *masendeshita*.

4.2.3 Inferring missing inflection paradigms

Most verbs and adjectives in the Wiktionary dump contain references to inflection table templates, but not all do. There are 4.171 verbs whose POS tags are followed by the name of their inflection paradigm, but also 1.514 where this information is missing. As for adjectives, 2.819 include inflection information and only 155 lack it.

Among the entries without inflection paradigms are words for which inflection information is given in their POS tag. For example, 贖う *agau*/*aganau* ("to atone for") has the POS tag `V1`, but does not refer to an inflection paradigm. From the POS tag and the ending of the lemma, we can infer that this verb belongs to the *go-u* inflection class, a subcategory of the godan class (see also Table 1).

In such cases where the combination of inflection information in the POS tag and the word form itself lets us infer the actual inflection class with high confidence, we attempt to do so. For this reason, we do not infer inflection paradigms of adjectives, since their POS tag includes

[16] For the non-base forms, it would be more accurate to refer to this as a sequence of suffixes (the stem-forming suffix followed by one or more inflectional suffixes).

no additional information. We also do not generate inflections for nidan or yodan verbs, since they are obsolete.

Before associating a token with an inflection class, we ensure that it meets the morphological criteria for belonging to that group:

- Godan verbs need to end with an *u*-vowel syllable. To get the correct Wiktionary inflection table template, we get the Latin transliteration of the final syllable (e.g. a verb ending in -む *mu* is assigned the template *go-mu*).
  - We do not assign the category *go-ou* to any verbs for which it is not already the associated paradigm. This class is a special subgroup of *go-u* that is used for some verbs in the Kansai dialect (Martin, 1975, p. 476). Because a verb ending in *-ou* could belong to either inflection paradigm, but there are significantly more *go-u* entries than *go-ou* entries in the Wiktionary dump[17], we assign *go-u* to all verbs ending in *-ou*.
- Ichidan verbs need to end in *-eru* or *-iru*. Note that there are also godan verbs with such endings.
- Type-III verbs belonging to the *zuru* class end with -ずる *zuru*, since the inflection base-forming suffixes for these verbs also need to begin with voiced consonants. The verb classification on Wiktionary also largely distinguishes between two groups of composite する *suru* verbs that are derived from single-kanji nouns and the group that is derived from nouns consisting of multiple kanji or foreign words. Generally, verbs from the latter group are assigned the inflection paradigm *suru*. Single-kanji する verbs whose pronunciation ends with -っする *ssuru* are in the *suru-tsu* group, whereas most of the other single-kanji する verbs belong to *suru-i-ku*[18]. When assigning inflection table templates to verbs without inflection information, we follow this structure.

If the POS tag of a verb does not indicate its verb group, but ends with -する *suru* (or -ずる *zuru*), we try to assign it a type-III class, as detailed in the previous step. Otherwise, if it cannot be an ichidan verb, we assign it a godan class, like explained above.

4.2.4 Applying the templates

When performing the above-mentioned (section 4.1) conversion entries of the Wiktionary dump into `Token` objects and we come across a tokens with inflection paradigm information, we generate all inflections from the corresponding template. We also save the lemma token as an `InflectableToken` instead of a plain `Token`, in order that it can contain references to its inflected forms.

In general, we use the templates by removing the final syllable to get the root of the word and appending the suffix given in the template to create a new `InflectedToken` that contains a reference to the inflection and the base form token in addition to the usual token properties. The suffix from the template can be empty, as is the case for the imperfective and continuative bases for ichidan verbs.

There are a few exceptions to this:

- The verb する *suru* (or its voiced version ずる *zuru*) is sometimes attached to a foreign-language noun to derive a verb from it (McClain, 1981, p. 11). When dealing with such verbs, we remove the last two syllables, instead of just one, before appending the suffix from the inflection table templates. This applies for verbs with the Wikitionary inflection paradigms *suru*, *suru-i-ku*, *suru-tsu* and *zuru*.

[17] The Wiktionary dump includes 211 *go-u* verbs and two *go-ou* verbs.

[18] This distinction between different kinds of single-kanji する verbs generally matches the description in Kageyama and Kishimoto (2016, pp. 96–98), although Kageyama and Kishimoto also group verbs ending in -んする *nsuru* with the -っする *ssuru* verbs, based on their negative inflection.

- For some verbs, the pronunciation of the root changes for specific bases or inflections. In these cases, the templates we created include the entire inflected forms as opposed to just the suffixes.
  - *aru*: The plain negative forms of 有る/ある *aru* are based on another root, the verbal adjective 無い/ない *nai* (McClain, 1981, p. 9). Because that root uses another kanji character, our template includes the kanji. To get the pronunciation of an inflected form of 有る, we use the token form, but replace the kanji with the corresponding hiragana character (i.e., 有 is replaced with あ, and 無 with な).
  - *kuru*: A special property of the verb 来る *kuru* is that the vowel quality of 来 changes for some of the inflections–this kanji is represented by different kana when transcribing its pronunciation. Therefore, we included the kana versions of this syllable in the template. To get the kanji forms, we replace the first syllable from the template entry with 来. There are also composite verbs ending in -来る (e.g. 連れて来る *tsuretekuru* ("to bring someone")). For these, we generate the inflected form of 来る and its pronunciation, and append it to the first part of the verb (e.g. 連れて-).
  - *suru-indep*: 為る/する *suru* is also conjugated irregularly when occurring on its own. Similarly to 来る *kuru*, we included the inflected forms in their entirety in the template, and we used hiragana. If necessary, we also restore the kanji character.
- The lemmas of the nominal adjectives do not have endings that need to be removed before adding the inflectional suffixes from their templates. This concerns entries with the inflection paradigms *na*, *nari* and *tari*.

Some of the templates contain suffixes with optional syllables (e.g. the conditional suffix in *na* is なら（ば） *nara(ba)*). In these cases, we generate `InflectedTokens` for both versions (e.g. なら and ならば).

4.3 Performing the initialization only once on server startup

Previously during the course, we performed resource-related activities, e.g. initializing data structures from external files, only within the service implementations on the server side. However, this is not ideal for our tool, since the process would be needlessly repeated on each client request. If the initialization process takes a fair amount of time and resources, the server would soon be overwhelmed by having to do the same grunt work every time a request is received, and the user experience would be bad since the user would have to wait for a few seconds every time before the results to their query are received.

Initialization of resources on server startup is a common scenario, and there are actually already existing solutions that address this need.[19] We decided to choose the `ServletContextListener` approach, which is detailed below:

4.3.1 Loading the resource in `Listener` class

First, we simply created a class called `Listener` in the `server` package. The key is to let the class implement `ServletContextListener`.

After creating the class, we needed to implement the method `public void contextInitialized(ServletContextEvent sce)` in the class, which will be executed when the server is started. We load the Wiktionary dump and perform all preprocessing steps in this method. As a result, we obtain the `Multimap` of Tokens, ready to be used in the queries.

[19] An answer on Stackoverflow lists them in detail: `https://stackoverflow.com/questions/3468150/using-special-auto-start-servlet-to-initialize-on-startup-and-share-application/3468317#3468317`

The key to reusing an object later is to store this object in the `ServletContext` with the method `setAttribute`.

4.3.2 Retrieving the resource in `LookupServiceImpl` class

After the object is stored in `ServletContext`, it can be retrieved again in any other servlet, in our case the `LookupService` servlet, where we retrieve the object whenever a user performs a lookup. This avoids the need to initialize the Wiktionary resource every time. Since the tokens are already stored in the server memory as an object, they can be retrieved instantly.

4.3.3 Registering the listener

Finally, it is necessary to register the listener in the `web.xml` file, similarly to how we registered the servlets. Note that it has to be registered before the `<servlet>` registration.

5 Processing User Input

Upon pressing the "Submit" button, the input is sent from the client to the server, where the server first uses the Kuromoji tokenizer (Atilika, 2018), which returns a list of tokens which are specified in the IPAdic (Asahara and Matsumoto, 2003) format. Using this list of IPAdic tokens as the basis, we perform lookups on the `Multimap` of `Token`s created from the Wiktionary dump. The lookup is necessary since the ipadic tokens differ from the `Token` we defined in certain aspects:

- The POS tags are specified in Japanese. The 4-level classification scheme used for the POS tags (Asahara and Matsumoto, 2003) is also different from the one used in Wiktionary (Wikimedia Foundation, 2018). A mapping and conversion between the POS tags is needed. For example, the POS tag CNT (counter words) in Wiktionary is classified as a sub-sub category of noun, as 名詞 - 接尾 - 助数詞, under the IPADIC scheme.
- The Kuromoji tokenizer always separates an inflected word into its base form and its inflectional ending. However, from the Wiktionary dump and the inflection templates, the tokens are generated in a way such that an inflected token is counted as a whole. Therefore, whenever a token from the Kuromoji outputs is marked as an inflection form, the search algorithm continues to look at the next token, until the token is no longer marked as special. Then, it tries to search for the match from the `Token` multimap by combining the several tokens marked as special, instead of using only one token to perform the lookup.

After the list of `Token`s is generated, it undergoes a sorting process so that the `Token` most likely to be the matching one gets put first. We achieve this by sorting the list on three levels: Primarily, we try to match the POS tag; secondarily, the pronunciation. Additionally, we prefer inflected tokens over uninflected ones to ensure that inflection information is also shown for the verb forms that are identical to the lemma.

6 User Interface

For the user interfacing styling, we use Bootstrap (Twitter, 2018) version 4.

6.1 Glosses pop-up

Since it is possible for a word to have multiple translation entries, we decided to show only one of these glosses in the results table. The rest of the glosses are displayed via a pop-up, which is displayed when the user clicks on the displayed gloss, as shown in Figure 2.

The gloss that is shown is the first gloss in the glosses list, after removing special entries such as the [archaic] ones (as described in section 4.1). For out-of-vocabulary tokens, we add a default gloss "[out-of-vocabulary]" or, in the case of punctuation marks, "[punctuation mark]".

The pop-up is a `UIBinder` widget. At the time when the results are generated, the widget is created and dynamically filled with a Bootstrap `list-group` class, where each gloss constitutes one `list-group-item`.

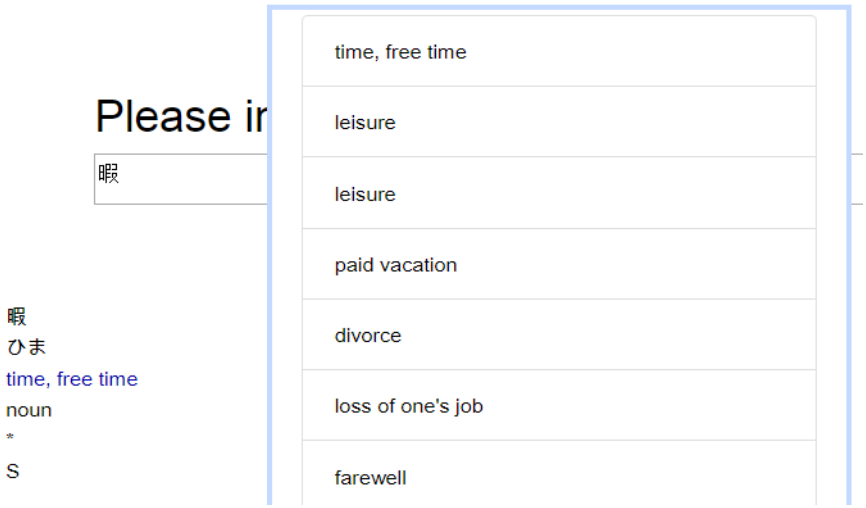

Figure 2: Glosses pop-up

6.2 Words with multiple entries

Furthermore, some words can have multiple entries with different POS tags and/or different pronunciations in the Wiktionary dump. Figure 3 gives an example.

```
破格　はかく　A[na]　1) exceptional, special, above average 2) ungrammatical, poor
(‘‘in terms of writing’’)
破格　はかく　N　1) exceptionalness 2) solecism
```

Figure 3: Words with multiple entries in the Wiktionary dump

Even though the Kuromoji tokenizer is already highly accurate when performing word segmentation and trying to determine the POS tag of each token, sometimes it can still err and infer the wrong POS tag, thus leading to our app showing the wrong meaning. In such cases, it is useful to show the alternative tokens as well, so that the users still get a chance to find the correct token and understand the original sentence.

This is implemented within the glosses pop-up mentioned in the previous section. Whenever the server-side code finds multiple matches for a token, it will put the list of tokens through a method `sortTokens`, before returning the results to the client. The client side displays the token that ranked first in the sorted results in the table by default, while appending the rest

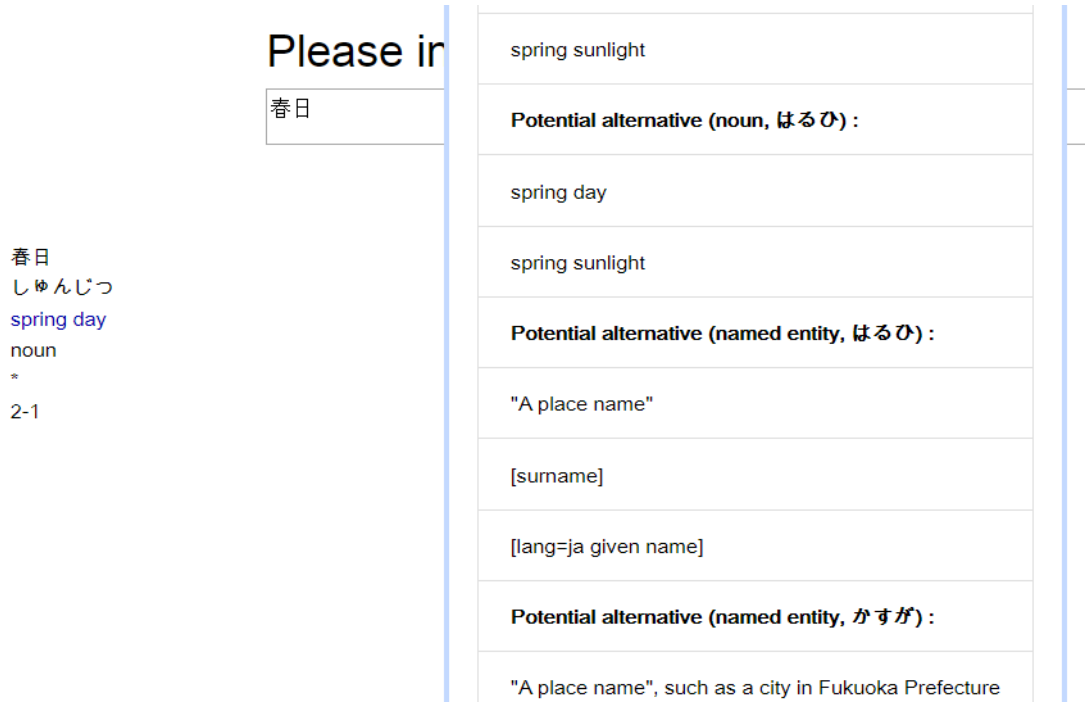

Figure 4: Excerpt from a multi-token pop-up

of the tokens to the list displayed by the glosses pop-up. The POS tags and pronunciations of the alternative matches are displayed together with their gloss entries, as shown in Figure 4.

6.3 Inflection table pop-up

For an inflected word, besides showing the inflection information for this particular form in the results table, we also provide a full list of possible inflections for the base form in the format of a pop-up widget, similar to the glosses pop-up mentioned above.

For this purpose, we created two additional classes, `InflectedToken` and `InflectableToken`, which extend the `Token` class.

- Each `InflectedToken` contains a field `lemmaToken` of type `InflectableToken`, which is the base token.
- Each `InflectableToken` contains a field `inflectedForms` of type `ArrayList<InflectedToken>`. This list contains all the inflected forms generated from this base token.

When displaying the results table, if the client side finds the type of a returned token to be `InflectedToken`, it displays its inflection information. In addition, it iterates the `inflectedForms` list from the lemma of this token to retrieve all available inflected forms of this token. Similarly to the glosses pop-up, all inflected forms as well as their inflection names are displayed in a pop-up table if the user clicks on the inflection information row. It is implemented with an HTML `<table>` element and the Bootstrap class `.table`. Figure 5 gives an example.

6.4 Kanji difficulty information

The results table contains a final row that contains the kanji difficulty information that is described in section 4.1.1. Figure 6 shows an example. From the last row of the table, we can

Figure 5: Excerpt from an inflection table pop-up

see that the character 寿 is taught in secondary school (i.e., after grade 6); the character 司 is taught in the fourth grade, the characters 食, 計, 算, 言, and 語 are all taught in the second grade, while the character 学 is already taught in the 1st grade. The non-kanji characters receive an asterisk as their corresponding outputs.

Figure 6: Kanji difficulty ratings

7 Division of Tasks

7.1 Xiang Ji

- Frontend/client-side coding (Input box, results table, pop-up widgets etc.) (section 6)
- Function for processing glosses (convert the glosses string into individual entries) (section 4.1)
- Function for token lookup (analyze and match the Kuromoji tokenizer outputs with preprocessed Wiktionary entries) (section 5)
- Code for `ServletContextListener` (initialization of resources only once on server startup) (section 4.3)
- Generating kanji difficulty ratings (section 4.1.1)

7.2 Verena Blaschke

- Conversion of the Wiktionary dump into `Token`s (including clean-up of translation strings and the generation of easy-to-understand POS information) (section 4.1, excluding 4.1.1)
- Researching inflection in Japanese and analyzing how it is performed in the Wiktionary project (section 4.2.1)
- Preparation of inflection table templates (where missing) (section 4.2.2)
- Inferring missing inflection information (section 4.2.3)
- Generation of inflected forms (section 4.2.4)
- Sorting Wiktionary tokens by how well they match a Kuromoji token (part of section 5)

8 Discussion

8.1 Comparison with the original proposal

The final project implementation accomplished the goals set out in the original project proposal. Figures 7 and 8 show the GUI in the project proposal and the actual project, respectively, given the example input この動物は日本語で何と言いますか。 *Kono doubutsu wa nihongo de nan to iimasu ka.* ("What do you call this animal in Japanese?"). We also implemented additional functionalities mentioned at the end of the proposal, e.g. initialization of resources on server startup, providing full inflection tables and difficulty ratings.

この	動物	は	日本語	で	何	と	言います	か	。
この	どうぶつ	は	にほんご	で	なに/なん	と	いいます	か	
this (near the speaker)	animal	topic marker	Japanese	at, by, with	what	with, if, and	to say, to call	interrogative particle	
							formal inflection of 言う		

Figure 7: The GUI sketch presented in the project proposal

One thing that is slightly different from originally described in the proposal is the way in which difficulty rating of a word is retrieved. Originally we considered querying the website

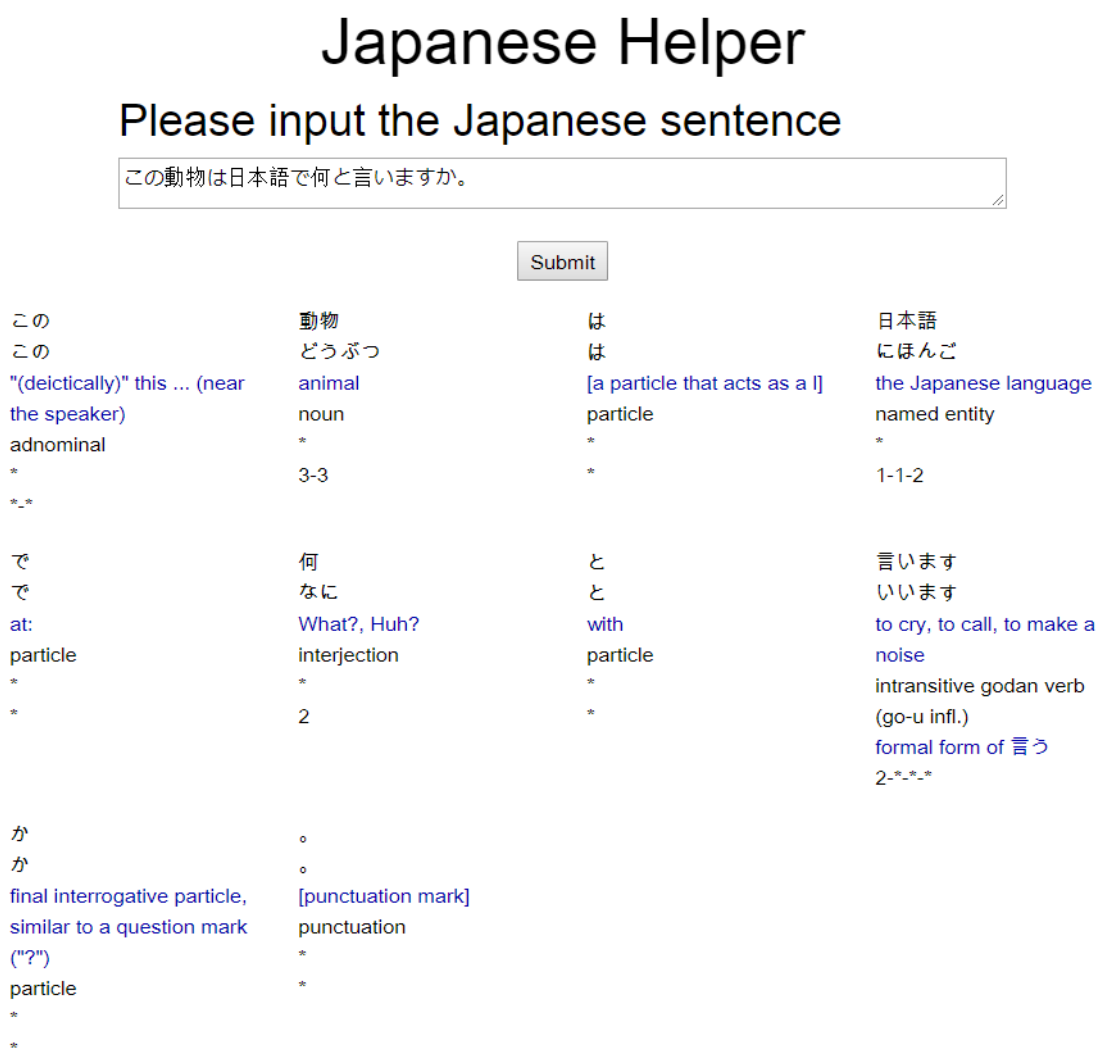

Figure 8: A screenshot of the actual GUI

`http://jisho.org/` for the difficulty rating of entire lemmas. However, their API currently only allows querying individual words. This means getting information for a full sentence, not to mention a full paragraph, would require sending multiple separate queries for the individual words in the sentence, which is impractical. Therefore, we took an alternative approach, which is to directly reference the list of common kanji and give the difficulty ratings of the words based on that.

In the project proposal, we considered displaying special information for certain tokens, akin to the so-called "usage notes" on Wiktionary in case of matches with several tokens, especially when several different potential pronunciations are available. Because parts of this information is sometimes contained in the translation lists in the Wiktionary dump, but without a predictable structure (e.g. sometimes in between parenthesis or brackets, sometimes directly as part of the translations, sometimes inside HTML comments), we decided to leave it in the glosses. We also decided against getting these usage notes by querying Wiktionary itself since that defeats the purpose of using a Wiktionary dump. Additionally, as with the difficulty ratings, it would slow down our tool. We are confident that the combination of glosses, pronunciations, part-of-speech information, and inflection information is helpful on its own in many of these cases. Furthermore, sorting tokens in these cases by pronunciation relative to

Kuromoji's proposed pronunciation also helps disambiguation.

8.2 Shortcomings and potential future improvements

Kuromoji is already a very accurate tokenizer and POS tagger. However, it sometimes still produces erroneous results, as all such tools currently do.

8.2.1 Compound nouns

Some compound nouns/proper nouns are listed in Wiktionary but not recognized by Kuromoji, and would therefore be segmented into multiple parts. For example, 時計回り *tokeimawari* ("clockwise movement") would be segmented as 時計 *tokei* ("a clock, a watch, a timepiece, a timekeeper") and 回り *mawari* ("rotation, circulation"). We considered trying out every possible combination of consecutive segmentation outputs which are assigned the POS "noun". However, that would be costly when the input is long, and might produce erroneous results, especially if other words are mistakenly tagged as "noun" by the tokenizer[20]. Therefore, we did not include it in the end. To a human user however, the re-combination of, e.g., "clock" and "rotation" to "clockwise movement" within the context of the sentence is not difficult.

8.2.2 Context-dependent meanings or pronunciations

Some words have context-dependent pronunciations. For example, "何" is usually read as なに, but before だ, で, の, なり or counters, it is read as なん. However, both the tokenizer output and the tokens created from Wiktionary dump would always give its pronunciation as "なに" no matter under what circumstances.

The only solution for this seems to be to hard-code exceptions into the program, which can be collected given enough time. However, as this is a semester project, we did npt seek out those exceptions. Wiktionary also has usage notes for some of the exception scenarios, but they are relatively hard to scrap and were not included in the Wiktionary dump from which we generate our tokens. Therefore, we were not able to show such information.

References

Asahara, M. and Y. Matsumoto (2003). *IPADIC version 2.7.0 User's Manual*. Nara Institute of Science and Technology. `https://ja.osdn.net/projects/ipadic/docs/ipadic-2.7.0-manual-en.pdf/en/1/ipadic-2.7.0-manual-en.pdf.pdf`.

Atilika (2018). Kuromoji. `http://atilika.com/en/kuromoji`.

Hamano, S. and T. Tsujioka (2011). *Basic Japanese: A Grammar and Workbook*. Routledge.

Iwasaki, S. (2013). *Japanese: Revised Edition*, Volume 17 of *London Oriental and African Language Library*. John Benjamins Publishing Company.

Kageyama, T. and H. Kishimoto (Eds.) (2016). *Handbook of Japanese Lexicon and Word Formation*, Volume 3 of *Handbooks of Japanese Language and Linguistics*. Walter de Gruyter.

Komai, A. (1979). *A Grammar of Classical Japanese*. Culver Pub.

Martin, S. E. (1975). *A Reference Grammar of Japanese*. Yale University Press.

[20]Note that alternative POS tags for 回り in the Wiktionary dump are "suffix" and "verb" (in an inflected form). Depending of the context of 時計回り within a sentence or discussion, the non-noun entries might be called for.

McClain, Y. M. (1981). *Handbook of Modern Japanese Grammar: Including Lists of Words and Expressions with English Equivalents for Reading Aid*. The Hokuseido Press.

Shibatani, M., S. Miyagawa, and H. Noda (Eds.) (2017). *Handbook of Japanese Syntax*, Volume 4 of *Handbooks of Japanese Language and Linguistics*. Walter de Gruyter.

Stalph, J., I. Hijiya-Kirschnereit, W. Schlecht, and K. Ueda (Eds.) (2009). *Großes japanisch-deutsches Wörterbuch*, Volume 1 (A–I) of *Großes japanisch-deutsches Wörterbuch*. Munich: Iudicium.

Twitter (2018). Bootstrap. `http://getbootstrap.com`.

Wikimedia Foundation (2018). Wiktionary:About Japanese. `https://en.wiktionary.org/wiki/Wiktionary:About_Japanese`.