# Setting up Japanese NLP with spaCy and MeCab

*2019-10-04T15:59:42+09:00*

*This post is part of a collection on Natural Language Processing.*

Modern Japanese NLP work relies on a number of tools that, while mature and effective, aren't necessarily well documented or described in once place, particularly in English. This post is a short guide to getting spaCy set up to work with Japanese using MeCab and UniDic for tokenization.

---

**Update:** Due to issues with mecab-python3, in late 2019 spaCy switched to fugashi, a Cython wrapper for MeCab I wrote, for Japanese support. On April 15, 2020, I released a version of fugashi that, in addition to having wheels for major platforms, allows you to install UniDic via PyPI, so you don't need to follow any of the steps below. That said, if you do need to install MeCab for some reason, this is still a good guide.

---

First, how do you process Japanese text? One major difference between Japanese and many other languages is the lack of spaces. This means that tokenization, often a trivial step in English NLP, is a significant task all by itself. Typically tokenization in Japanese is modeled as a joint task with part-of-speech tagging. While there are many promising tokenizers in development in 2019, currently the best Japanese tokenizer is still MeCab; you can also read the paper that introduced its lattice-based tokenization algorithm.

旧芝離宮恩賜庭園

# Installing MeCab

One problem with MeCab is that there hasn't been a release in years and it's almost entirely unmaintained. Many Linux distributions have packages for MeCab, but since they're often out of date I strongly recommend installing directly from the source on Github.

```
# install iconv using your OS package manager
git clone git@github.com:taku910/mecab.git
cd mecab/mecab
./configure --enable-utf8-only && make
sudo make install
```

Using `--enable-utf8-only` is optional here, but unless you really need to work with SJIS or EUCJP directly it'll make your life easier.

MeCab comes with two dictionaries, IPADic and JumanDic. These are both long abandoned, so you'll want to install UniDic, the only supported dictionary for Japanese Universal Dependencies.

# Installing UniDic

UniDic has its own problems - while it's actively maintained,

recent changes have added many entries of dubious utility to the dictionary, drastically increasing its size. If you're just getting started using a slightly older version is fine, so go to the UniDic downloads page and download the file named `unidic-mecab-2.1.2_src.zip`. After that:

```
unzip unidic-mecab-2.1.2_src.zip
cd unidic-mecab-2.1.2_src
./configure && make
sudo make install
```

Now you'll have to update your `mecabrc` file to use UniDic. Typically this file will be in `/usr/local/etc/mecab`, though depending on your OS it may be in a different location. Once you find the file, change the `dicdir` to the path where UniDic was installed.

At this point you can check that everything is working by using `mecab -D` to dump dictionary info, which should show you're using UniDic. If that's OK type `mecab` and it will read text to tokenize from `stdin`. If you paste 国立国会図書館 you should get output like this:

```
国立    コクリツ        コクリツ        国立    名詞-普通名詞-一般
国会    コッカイ        コッカイ        国会    名詞-普通名詞-一般
図書    トショ  トショ  図書    名詞-普通名詞-一般
館      カン    カン    館      接尾辞-名詞的-一般
EOS
```



# Python & spaCy

Now that MeCab and UniDic are properly installed you're finally ready to move on to Python. To use MeCab from Python you'll need the `mecab-python3` wrapper package. Recent versions of this have bugs, so you'll need to install an older version.

```
pip install mecab-python3==0.7
```

After this you can finally install spaCy and play with Japanese support. You can install spaCy without any special options and it'll just work. Here's a quick test:

```
import spacy
```

```
ja = spacy.blank('ja')
for word in ja('日本語ですよ'):
    print(word, word.lemma_, word.tag_, word.pos_)
```

And you're ready to go!

Currently this process works, but honestly it's a mess. I'm working on making things easier; my goal is that in 2020 all of the above can be replaced with a single `pip install` command. If you're interested in my progress you can follow me on Twitter or check out the Japanese model issue for spaCy. Ψ

Dampfkraft is the home page of Paul McCann, who lives near Tokyo Tower with a jade tree. You can follow him on Twitter here or check Cotonoha to hire him for NLP work.