ISMLA 17/18 Exercise 02: Recognizing Foreign Names in Chinese

1 Changes to Project Setup

- Use the new `cmnengdictphon.jar` instead of `cmnengdict.jar`.
- You can reuse the `TokenCMN` and the test data from the last exercise.

2 Implementation Step

1. Define two new subtypes of `TokenCMN` without any additional features: `MeaningfulTokenCMN` will be used to annotate the Chinese words recognized by the greedy tokenizer, and `PhonChunkCMN` to mark spans which might represent foreign names.
2. Adapt your greedy tokenizer to work on a `CmnEngDictionaryWithPhon` instance, and to ignore all `SimpleDictionaryEntry` with `cat` value `PHON`. Instead of general `TokenCMN`s, it should now create `MeaningfulTokenCMN` annotations.
3. Write a new primitive analysis engine that annotates as `PhonChunkCMN` instances consecutive chunks (length ≥ 2) of Chinese syllables which have a dictionary entry with `cat` value `PHON`. The `cat` value should be `PHON`, the gloss value should be concatenated from the first glosses provided for each syllable (a best guess of Latin equivalents) and put into square brackets, and the Pinyin value should be just the concatenation of the syllables' Pinyin values. Note: ≥ 2 means that isolated `PHON` symbols are ignored!
4. Write another primitive analysis engine which reworks the annotations generated by the previous two annotators. For each `PhonChunkCMN`, use the `AnnotationIndex<MeaningfulTokenCMN>.subiterator(chunk)` method to check whether it includes a `MeaningfulTokenCMN` of length ≥ 2. If it does, remove the `PhonChunkCMN` annotation; otherwise, remove all the meaningful subtokens, so that only the `PhonChunkCMN` remains annotated.
5. Write a CASConsumer which prints the `TokenCMN` annotations of each document into a separate TSV file. The output format should include

the following columns: orthography, pinyin, category, and gloss. The files should be generated in a directory that can be configured by a parameter.

6. Define an aggregate analysis engine which chains together the four components you built so far in a pipeline. Execute it on the directory with the five Wikipedia snippets from last time.

7. Inspect the gloss output and try how far they help you to understand the snippets. If you cannot interpret the glosses, running the snippets through Google Translate will help you to make sense of them. To see whether the named entities were correctly recognized, look some of them up in the Chinese Wikipedia (`zh.wikipedia.org`) until you hit some entities you cannot find in this way. There is a common pattern to these examples which has something to do with multi-segment tokens that overlap the boundaries of the `PhonChunkCMN`s. What is the problem? How could the algorithm be refined to handle this case?

3 Submitting your result

Come to one of us **as a group** to show and explain your output and code:

- Björn's office on Fri, November 17, 2-3 pm
- before or after the lecture on Mon, November 20 (half an hour each)
- Johannes' office on Tue, November 21, 1-2 pm
- by appointment **before** the next exercise session (November 22)