Review Interview Study Subject01

**Demographics**

**Title** Researcher and Software Engineer

**Company** Company 1

**Education** Masters of CS

**Experience in data analysis** 10 years (all with text)

**Experience in review analysis** 2 years

**Gender** Male

## Summary

**Project**:

Attribute extraction.

Goal: doing entity + opinion extraction over reviews.

**Downstream applications**:

subjective search, SQL-queryable version of an attribute extraction algorithm.

**Pipeline**:

Extractions → hisograms (aggregated extraction results per entity)

Find phrases from reviews that are relevant to a given search term

**Data**:

Booking.com, yelp (O(10k)), and proprietary data (~3 million, English and 95+% Japanese) reviews. English and Japanese.

Accessed via google cloud and BigQuery, and copied into PostGres because interviewee is most comfortable with this method.

Yelp and booking.com are pretty clean (public datasets).

proprietary reviews were directly from production. Cleaning process: remove reviews for properties no longer working with the company (cleaning is based on meta-data, not raw text)

All reviews are parse-able by spacy and spacy-japan. Most of the errors are caused by the dependency parser (segmentation + POS tags are mostly correct). Interviewee relies on splitting sentences with multiple sub-sentences, rather than the dependency tree. This means losing some extractions that go over multiple sub-sentences but it removes some error propagation.

Unsupervised learning: he doesn’t use training data. The extraction is based on rules and a schema (defining words of interest for each topic).

**Evaluation**: has dev-set (20) and test-set (20) for tuning, labeled himself.

Rule-based is enough? → unsure. It’s more interpretable than the ML approaches.

Spends most time

* using the dev set to tune the schema (mostly adding new seed words and expansion parameters)
* Finding bugs in the extractor code

Pain points:

* Labeling test set and dev set. Labels are subjective, many ways to do it, and sometimes makes mistakes. He feels confident in his labels (no 2nd opinion), but has to revisit labels many times as the opinion definition evolves.
* Organizing the pipeline. New reviews come in every day, so the data-processing pipeline is chaotic.
* Doesn’t store all his SQL, just runs it from the command line and it’s hard to remember/organize.

**Features/tools wish list**

A better way to organize the pipeline to handle new data from the proprietary source

**What tools and languages do you use?**

Spacy, ginza, python, postgres, sudachipy

Wrote some bash script to run extractions in parallel.

**How do you present your results? (To whom?)**

Mostly share csv files, postgres tables, or json.

Some figures when he shares with people outside of the project