

An Investigation into the uses of Factotum and extending its capabilities

By Claire Abu-Hakima

ABSTRACT (4 sentences)

Factotum old

Improved/updated

Lack of ling db, would be super useful

With rudiment data set from wiki, showed its utility

I. INTRODUCTION (1 pg)

The job of a researcher requires one to gather all possible relevant data in hopes of supporting an already stated hypothesis or discovering a new one. However, the task of organizing and sifting through all the data is often tedious and identifying unknown patterns is not always obvious. And so, it was in 1993 that Robert Uzgalis created Factotum, a software tool designed to help with these very issues.

The original program, described in “Factotum 90: A Software Assisted Method to Describe and Validate Data,” would allow researchers to uniformly enter their data, and suggest patterns and connections between given facts, creating a sort of taxonomy for the different data. Additionally, Uzgalis made a singular format for entering data allowing it to be language independent, so that facts in any natural language may be used with Factotum.

Another advantage to using Factotum in addition to aiding the researcher store their collected data is the forced formalism of the data through the use of Factotum. The format in which the researcher must shape the data requires him to further think critically of their work and to come closer to realizing what it is they wish to demonstrate through this precision as well as what the data is actually showing them.

Over the years progress in the development of Factotum has been minimal and the software has not been used how it was originally envisioned. However now, some seventeen years later, Factotum is being updated into Python from C with some parts even being rewritten and along with a new interesting idea for its use. I’ve rewritten the parser for the vocabulary of the data (a set of rules designating the format of the data), which makes sure that the format of the vocabulary rules themselves is correct, as well as the fact checker, which uses the generated vocabulary to check if the entered facts to adhere to those specified formats. Additionally, I created a specific dataset for Factotum, a dataset that was in fact the main driving force of the project.

As a student of Linguistics, I’ve always found the lack of a single unified and legitimate resource or database for the world’s languages to be surprising and, at times, frustrating. And so not only would Factotum provide a great format for organizing (mainly meta-) data on the world’s languages, but could also be used to demonstrate the connections between different language families, unforeseen ties to be further

explored, as well as gaps in knowledge. A Factotum database with collaborators from different experts on languages would be not only an excellent source for researchers just seeking essential meta data, but also a way to preserve and easily spread this knowledge.

However, given a lack of experts for now, I created a rudimentary data set made up of facts pulled from articles and mainly the information boxes on the Wikipedia language pages to demonstrate the potential that Factotum holds; I will give further details on what I learned from this collection of data, as well as what I learned was possible to do with it in Factotum itself.

II. THE PROBLEM (1 page) / MY IDEA (2 pgs)

In this section I will now discuss in further detail what exactly it is I am trying to do with Factotum as well as my idea for a linguistic database.

As a student myself, I always found myself researching different languages online, gathering data from different books and papers, and frustrated at the lack of a single, qualified data base containing (at the very least) meta data for all the languages. I say qualified because presently there exists a sort of database of this nature: Wikipedia.

IV. THE DETAILS (5 pgs)

Now I will discuss the more intensive details of the workings of Factotum, how I wrote and used the vocabulary parser and fact checker, how I created my Lingdata dataset and what results did I get when running the data through Factotum.

HOW FACTOTUM WORKS

Though there is not much documentation on Factotum except for the original paper, I will attempt to give a clear description based upon it as well as from what I have asked and learned from Uzgalis.

- TERMS: marker, alias, entitites, subject, predicate, object, citation

- FACTS: what are facts, entering data/data format, automatic or manual

- VOCABULARY

The vocabulary is contained in a separate file ending in '.v' which contains rules that dictate the format of facts in the .f file. Though the vocabulary may be generated automatically using the mkvocab module (provided by Uzgalis), it can be manually altered (or even created manually) to provide more accurate descriptions for the facts, though it is usually better to protocol to check for errors in the fact file first before

modifying the vocabulary. Human error is more likely in the fact file especially if the vocabulary has been automatically generated.

Additionally, the vocabulary itself has it's own format it must adhere to, which is confirmed by the vocabulary parser (predpar.py) that I wrote and will discuss in greater detail later on.

-predicates, types of rules, what does vocabulary signify, automatic or manual

VOCAB PARSER

One of the main bits of code that I wrote for Factotum was the vocabulary parser, which is contained in the module prepar.py. The ultimate goal of this parser is to check whether the rules contained within the given .v file adheres to the required format of vocabulary rules.

I accomplish this by first translating the designated format of the rules into a grammar which I represent using a dictionary called 'vocab_grammar', where the keys of the dictionary are the non-terminal symbols, and entries for each key represents the right-hand side (RHS) of the grammar rule with each token an item in a list, and where there are multiple RHS's, (for example Pred goes to := Phrase and -= Phrase, etc.), the entry is represented as a list of lists, so that our previous example would be like 'Pred' : [[':=', 'Phrase'], ['-=', 'Phrase']] . This is set as a global variable so that I can access it more readily when doing the parse.

Next I had to get the vocabulary rules from within the file into the appropriate format. And using the helper function 'go_thru_file' I use a similar method as used in the factotum_lexer module, where I read in the file line by line, identifying where the line is a continuation or a clean break and then storing it in a list of lists of the form [[subj, pred]...]. After that I iterate thru the rules in this list, and first tokenize the predicate using the 'tokenize_pred_sting' function, separating the string pred according to either symbols that are significant in vocabulary rules (which I represent in a regular expression) or just words (separated by white space).

FCHECK

LINGDATA

Another significant portion of this project lies not merely in the code for the parser, but also in the data itself. Here I will explain why I chose to create my dataset from Wikipedia, what select bits of data I chose to scrape off the Wikipedia language pages, how I managed to scrape that data and how I transformed that raw data into usable Factotum facts.

A. WHY WIKIPEDIA

As I mentioned earlier in the paper, as a student of Linguistics I was quite surprised and frustrated by the lack of a single unified and legitimate source containing information or at least meta-data on all of the world's languages. All that I was searching for was a quick reference to a language so that I could get an overall idea of what sort of language it was, the language family, and some other bits of information concerning either syntax or phonetics and related languages. However when searching for an academically sound source, I often would have to leaf through papers and papers to find the one little bit of information I was seeking, often times with varying opinions and unclear answers, or no answers at all. Time after time I found myself discouraged with and tired of searching for the simplest bit of information and having to do the run around, and so would often find myself turning to Wikipedia for reference despite what my professors had advised me.

While it may be the case that Wikipedia is not the most respected or complete source of information in academia, one cannot deny the sheer amount of crowd-sourced knowledge available in a single place. For essentially any language I would look up, there would be an entry, and though not always comprehensive or necessarily correct, it would provide the basic information I was seeking for back reference, while I could still go and look for more specific topics in scholarly journals and publications. Wikipedia provided the perfect platform to begin research and though I searched for a more reliable source, I continually found myself coming back to it.

So as I mentioned before, the potential for Factotum in the linguistics community is quite great, and to demonstrate that, I used information pulled from Wikipedia for my test case to show how Factotum, even if used on crowd-sourced information, can yield some very useful and interesting results.

B. INFOBOX/WIKIPEDIA LANGUAGES

As I previously mentioned, during my studies I continually found myself using Wikipedia as a reference for mainly meta-data on languages. What I came to notice is that every language page in Wikipedia has as part of its template an 'Infobox' referred to as 'Infobox _language' containing the most rudimentary information about any given language (note: all Wikipedia pages referenced in this project are in English). So for my Wikipedia test case, I decided to scrape every language page containing the Infobox, and to grab all the information within the Infobox and turn it into Factotum facts.

The Infobox is structured with 3 main sections with three separate headings and subheadings within it (not all of which are required). The first section is required, its heading is the name of the language (e.g. French), along with its native name (e.g. Français). Within this section there are many subheadings (recall not all are required), including different types of subheadings depending on whether a given language is a natural language, constructed, or a sign language. The subheadings that pertain to

mostly natural languages (some for sign languages) include 'Pronunciation', 'Spoken in' or 'Signed in', 'Region', 'Extinct Language' or 'Language extinction', 'Total speakers' or 'Total signers', 'List of languages by number of native speakers' or 'Ranking', 'Language family', 'Standard forms', 'Dialects', and 'Writing system.' The subheadings that refer to constructed languages include 'Created by', 'Date founded', 'Setting and usage', 'Category (purpose)', 'Category (sources)', 'List of language regulators' or 'Regulated by.' Essentially the first section outlines the most basic facts about the language.

Next there is the optional second section, with the heading 'Official Status,' containing information about where the language is official and listing all the countries or domains where it is (subheading: 'Official language in'), listing where it is a minority language, (subheading: 'Recognised minority language in') and what body regulates the language (subheading: 'List of language regulators,' or 'Regulated by').

Finally there is the third section, 'Language codes', and while on the Wikipedia page itself there is a map highlighting the regions where the language is spoken in varying degrees (e.g. mother tongue, official language, second language, and minority), I cannot represent this information visually in Factotum, and the information is already contained in the first section and the 'Official status' section which I may more easily translate into Factotum facts. But the rest of the information in the section I can represent in Factotum, with the subheadings representing different language codes 'ISO 639-1', 'ISO 639-2', 'ISO 639-3', and 'Linguasphere Observatory' or 'Linguasphere.'

And so from these three sections is where I would pull my data for my test case. It should be noted however that given that I'm pulling the information off of Wikipedia, not all information is going to be accurate, there is going to be some controversy regarding some of the Infobox data, and I am limited in recording disputing interpretations just by what different language pages are available and what information is provided in the Infobox.

When I initially was creating my own data set sans Wikipedia, I was overwhelmed by all the design decisions I was forced to make concerning each language, but with the Infobox, I get a clean cut of each language without much difficulty. However, there are a few things would be useful to include in addition to the Infobox, which I will mention briefly later on.

C. WIKISCRIP.T.py / Scraping

- Scrape Template:Infobox_language for what links to here
- scrape the links page, and keep going from links page to links page
- after grabbing 50 links from ea page, go thru and parse to get the infobox
 - parse wiki pulls info box, puts into 3 sections, and then parses it into recognizable sub-headings and then writes facts based on those subheadings
- lists the ID of the page and the link for the archive of the page

D. Factotum fact form

E. what additional information could be added to infobox/data set

- word order

- what sounds present in language

- typology

- grammar (mood, tense, gender, case, etc)

#resulting vocabulary

resulting profile

RESULTS

MEASUREMENTS

V. RELATED WORKS (1-2 pgs)

VI. CONC AND FURTHER WORK (1/2 pg)

Sources