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
RESULTS
MEASUREMENTS

V. RELATED WORKS (1-2 pgs)

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

I. Introduction (1 pg)
   a. What is factotum
      i. Written by Uzgalis
      ii. Original development and purpose
         1. Provide data models for researchers
         2. Language independent
         3. Force to formalize facts
      iii. what ended up happening to it

   b. What I did with Factotum
      i. Want to demonstrate it's usefulness
         1. Given my interest/study of linguistics, want to demonstrate the potential use of factotum as a research tool
            a. Created a data set of about 20 languages with different language attributes
            b. All data in one place/data base (generally very spread out online)
            c. Possible new connections formed/recognized
      ii. Extend its capabilities/what worked on
         1. Brief intro to vocabulary parser
         2. Brief intro to fact checker

      iii. Possible improvements for factotum
         1. Vocabulary generation:
            a. Less of a cycle of 'massaging by hand' then running it through the software,
               i. Either: more/entirely automated
               ii. Or mainly human created (which was in fact my first impression)

II. How Factotum Works & What I worked on
   a. Vocabulary parser
      i. Describe what is a vocabulary
         1. Vocabulary can be thought of as a set of 'rules' by which the data must abide

2. But the vocabulary is in itself also a set of facts, since there are generating rules, implying rules, and since in general it give us information about the data (even if it is in terms of formatting)

ii. How is the vocabulary presented
    1. Several forms of the vocabulary: (brief & clarifying)
        a. If blocks/ “rule restrictions”
        b. Type trees
        c. Tags and strings
        d. Strings
        e. rule is
        f. phrase is
        g. attribute
        h. nattribute
        i. generation rule
        j. implication rule
    2. within these, there may be “objects” which refer entities
        a. clarify how they are presented label, type, token type, etc.

iii. PARSER
    1. What my parser did
        a. Dealt with the strings
            i. Why
            ii. How
        b. Sample code
        c. Point out where called upon existing factotum functions (eg Lex, entities etc)
        d. What design decisions I made
            i. E.g. how to hold the data
            ii. How to traverse it

b. FACT CHECKER
    i. What is the FactChecker
        1. Purpose – make sure facts obey the vocabulary rules (e.g. format) prescribed to them
        2. Necessity—don’t want fact that can’t fit the model
            a. Can adjust the rules/fact format and in return get better results
            b. Will force user to look at data and rules more intenselyl

    ii. Code
        1. My code samples
        2. Code Functionality
            a. Control flow
            b. “internal structure of data”

3. Design decisions I made
    a. Helper functions
    b. errors it chooses to show

III. Using Factotum –Linguistic Data

c. Why use Factotum?
    i. Possible linguistic data model
    ii. Collection of attributes all in one place
    iii. Linguistic knowledge currently quite spread out
    iv. No centralized data base/ source of information
    v. Perhaps patterns that have not yet been recognized will become apparent

d. Design Decisions
    i. What I chose to focus on
        1. Linguistics → languages
        2. Natural languages
        3. Written language
    ii. Which attributes – all available facts
        1. Language family (Genetic)
            a. Can make connections well
            b. Represented as type (< >)
        2. Alphabet
            a. Phrasal, with option of a –variant
        3. grammar
            a. cases?
            b. Gender?
            c. Tense
            d. Moods
            e. Pronouns
            f. Inflection of adj/nouns/verbs/etc
        4. Typology
            a. Word order
            b. Unusual syntactic structure
        5. Phonetics/phonology?
            a. Unusal features??
            b. Eg clicks
        6. Stats/numbers [need better sources if going to include]
            a. Number of total speakers
            b. where official lang
            c. dialects
            d. types of dialects – details?
    iii. what attributes to skip
        1. historical
            a. not doing a historical linguistics analysis

b. origins of writing systems etc interesting, but not our scope

2. political /religious aspects
    a. many nuances involved with this, would want to do it justice but don't have time to do thorough discussion/ analysis of politics
    b. only mention will be:
        i. names – for example if essentially same lang but different names/dialects then will provide aliases for them
        ii. geography
            1. if language was introduced to a region due to imperialism,
                a. will just be noted that language X is official/spoken in country Y
    c. examples
        i. cumbersome
        ii. complicated
        iii. need to veryify
        iv. can be added by experts in the different languages → potential growth

iv. Issues with attributes
    1. Competing theories,
        a. If there is a generally accepted theory, then that is the one that is noted
        b. if there are competing theories
            i. no clear front runner
            ii. so I've made the decision to mention both of the contending/leading theories
    2. Ambiguity
        a. Lang doesn't follow pattern all the time
        b. Lang has some deviations from attribute (e.g. word order)
v. Case Studies:
    1. Arabic
        a. How represented data
        b. What additional data decided to use
        c. What learned from that data
        d. What learned that only Factotum demonstrated
    2. Serbo-Croatian
        a. How represented data
        b. What additional data decided to use

c. What learned from that data
d. What learned that only factotum demonstrated

vi. Results of running data set with factotum
1. Excerpt/printout of data
2. How exactly I ran it
a. So that in future it will be easier to reproduce
3. Excerpt/print out of results
a. Description of what each section of results demonstrates
b. How do these results help us/ be useful to others
4. After running fact checker with the results,
a. How did I have to adjust vocabulary
i. why did we adjust it
b. How did I have to adjust facts
i. Why did I adjust them
ii. What results did that yield

IV. Improvements to Factotum
a. Vocabulary structure after mkvocab
b. Either more automation or less automation of vocabulary, combination makes things tricky
c.

V. Conclusion
a. What learned about factotum/parsing through data
b. What learned from linguistic data set
c. What learned about data models

Sources