ruft: A Python Package for Rule Generation from Raw Text

Olegs Verhodubs
oleg.verhodub@inbox.lv

Abstract. The abundance of electronic texts necessitates their processing. Knowledge extraction is the most natural method for processing electronic texts. The acquired knowledge can be used in both question-answering and knowledge-based systems. Therefore, we present ruft: a Python package that enables knowledge extraction from electronic texts. The package makes it possible to extract knowledge in the form of if..then rules from electronic text, processing one sentence at a time.

Keywords: Python, Natural Language Processing, Rule Generation, Expert Systems, Knowledge Acquisition

Code metadata

Current code version	0.4
Legal Code License	Apache License 2.0
Software code languages, tools, and services used	Python
Support email for questions	oleg.verhodub@inbox.lv

I. Motivation

There are a lot of texts in the Web, but the means for their intellectual processing are in short supply. There are various ways to process electronic texts. One can combine electronic texts from various sources into a single electronic text and then search it for what is needed. However, raw text typically contains a lot of secondary information, so researchers have developed a special format for capturing the essence of information through its structure. An ontology description language, such as OWL (Web Ontology Language) [1], is a format for describing information structure, and this language is part of the Semantic Web [2] concept. Electronic text prepared into a structure (i.e., described using an ontology) is much more amenable to processing, but the process of developing ontologies is labor-intensive. Another way to process electronic texts is to use large language models, but their use requires significant computational resources, even assuming their use in update mode.

In this work, we present ruft, that is, the prototype of the Python package for rule generation from raw text. Ruft prototype grew out of the need to reason on the basis of raw text [3]. The package can be used from simple Python scripts allowing developers to quickly and interactively experiment with rule generation from raw texts. Our package is built upon popular tool from Python's natural language processing ecosystem, such as nltk [4] package. The ruft package is available for installation from the Python Package Index (PyPi).

ruft package is useful for acquiring knowledge from raw text. Here, knowledge refers to rules in the form of If...Then statements. This is essential for knowledge-based systems, as well as for expert systems that can automatically populate their knowledge bases with rules generated from various electronic documents in the Web.

It is possible that other Python packages exist for generating rules from raw text. This issue has not been adequately explored, as the author has been exploring the idea of generating knowledge from any structure for a long time, and a package for generating rules from raw text is a logical outcome of previous research.

II. Software description

The prototype of the ruft package is implemented in Python. This prototype uses nltk package for natural language processing. That is, the ruft package prototype is designed to generate rules from raw text. To do this, the entire text must be broken down into sentences and then matched to rule templates. In turn, rule templates are defined using parts of speech.

The prototype of the ruft package is implemented as a ruft class, which contains several methods.

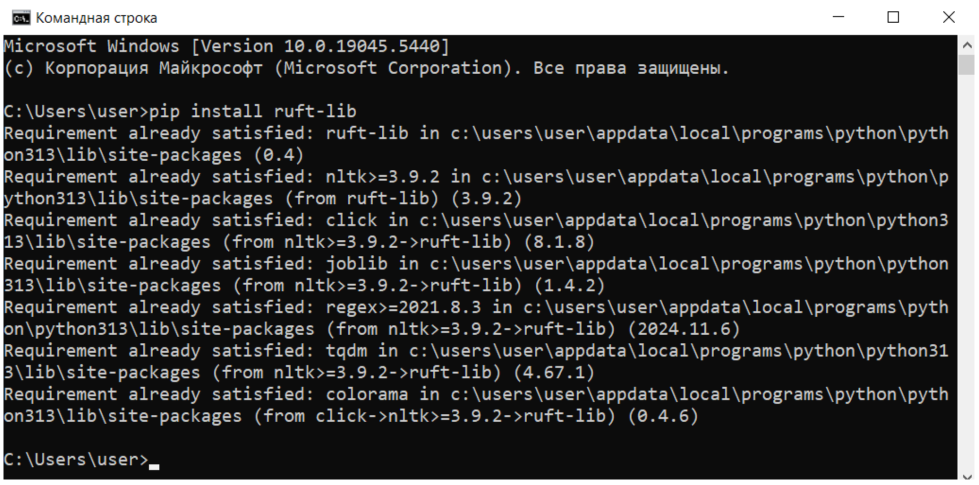
TABLE I. Methods of ruft class.

Method	Example of sentence	Generated rule
is_a_noun_rule(s)	An apple is a fruit	IF is a fruit THEN apple
is_a_adj_noun_rule(s)	A pear is a green fruit	IF is a green fruit THEN pear
is_adj_rule(s)	The oak is tall	IF is tall THEN oak
nnp_vbd_vbn_in_nns_rule(s)	Chess was invented by Indians	IF was invented by Indians THEN Chess
nn_consists_of_nn_and_nn_rule (s)	The meal consists of rice and meat	IF consists of rice and meat THEN meal
get_all_rules(s)	all sentences above	all rules above

Table 1 includes a list of the methods in the roof class and the types of rules they generate. Initially, in this prototype package, rules are generated in the classic format, i.e. in the If...Then form, as shown in the 3rd column of the table.

III. Examples of use

First of all, it is necessary to install ruft package to use it. To do this, we perform the standard steps, as with any other package, that is, we launch the command line (cmd.exe or powershell.exe) and execute the *pip install ruft-lib* command (Figure 1).



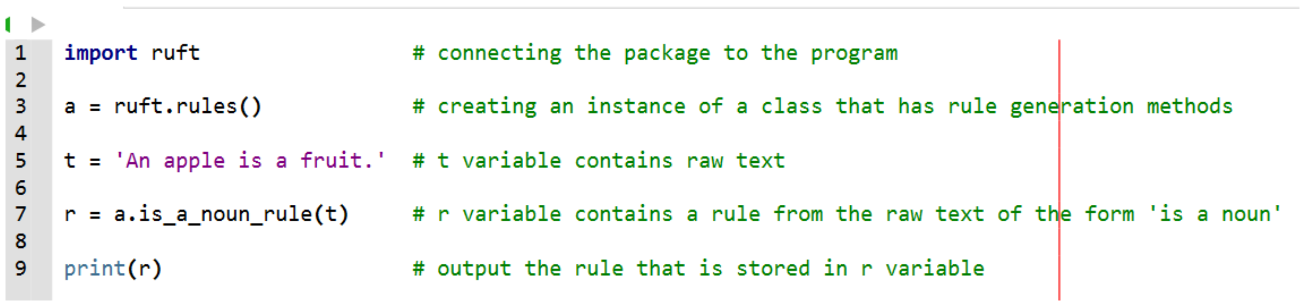
```
Microsoft Windows [Version 10.0.19045.5440]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\user>pip install ruft-lib
Requirement already satisfied: ruft-lib in c:\users\user\appdata\local\programs\python\pyth
on313\lib\site-packages (0.4)
Requirement already satisfied: nltk>=3.9.2 in c:\users\user\appdata\local\programs\python\p
ython313\lib\site-packages (from ruft-lib) (3.9.2)
Requirement already satisfied: click in c:\users\user\appdata\local\programs\python\python3
13\lib\site-packages (from nltk>=3.9.2->ruft-lib) (8.1.8)
Requirement already satisfied: joblib in c:\users\user\appdata\local\programs\python\python
313\lib\site-packages (from nltk>=3.9.2->ruft-lib) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\user\appdata\local\programs\pyth
on\python313\lib\site-packages (from nltk>=3.9.2->ruft-lib) (2024.11.6)
Requirement already satisfied: tqdm in c:\users\user\appdata\local\programs\python\python31
3\lib\site-packages (from nltk>=3.9.2->ruft-lib) (4.67.1)
Requirement already satisfied: colorama in c:\users\user\appdata\local\programs\python\pyth
on313\lib\site-packages (from click->nltk>=3.9.2->ruft-lib) (0.4.6)

C:\Users\user>_
```

Fig.1. Installing the ruft package.

Once the ruft package has been successfully installed, we can use it in our Python programs. To do this, launch Python or one of its shells (for example, Wing) and include the ruft package in the program using import, specifically import ruft. Next, to generate rules from raw text, you need to use one or more of the methods listed in Table 1. For example, as shown in Figure 2:



```python
import ruft                 # connecting the package to the program

a = ruft.rules()            # creating an instance of a class that has rule generation methods

t = 'An apple is a fruit.'  # t variable contains raw text

r = a.is_a_noun_rule(t)     # r variable contains a rule from the raw text of the form 'is a noun'

print(r)                    # output the rule that is stored in r variable
```

Fig.2. Using the functionality of the ruft package to generate rules.

Similarly, other methods can be used to generate rules from sentences of other types of text. All methods are listed in Table I.

IV. Perspectives

A prototype Python package for generating rules from raw text is presented in this paper. This prototype package implements the ability to generate five types of rules from raw text, with the ability to generate each rule individually or all at once. This prototype package has been published in the Python package repository located at PYPI.COM. Therefore, anyone can install it from this package repository and use it for any purpose.

This is a prototype package, not a full-fledged package. This is because the potential for rule generation is far from limited to the five rule types that can be generated from raw text. There are numerous unimplemented capabilities for generating other types of rules from raw text. Once at least some of these are implemented, and an optimal syntax for the generated rules is developed, we can begin to talk about a fully-fledged Python package for generating rules from raw text. All of this will be realized if there is interest from stakeholders and funding is available.

Initially, the task of generating rules from raw text arose during the development of the Keyword Search Engine Enriched by Expert System Features [5], but its implementation did not require the functionality to be packaged separately. However, implementing the task of generating rules from raw text could be useful to other developers, so the decision was made to develop a separate package for use by others. This is how the prototype of the ruft package was born.

This is not the first attempt of this kind: code for generating rules from OWL ontologies has already been developed and published (Rufron library [6]). It was implemented in Java. It was intended to develop this project to include the ability to generate rules from raw text, but the Python language appears more promising, so the ruft package was developed.

Acknowledgments

Gratitude is expressed to friends and family for their long-term moral and material support in the face of numerous illegal and discriminatory actions by the Latvian state, namely the Latvian police, the Latvian prosecutor's office, the

Latvian State Security Service, and the Riga Technical University, against the author in particular and against national minorities living in Latvia in general.

The illegal and discriminatory actions by the aforementioned Latvian institutions against the author can be proven by independent, qualified, legal, non-Latvian expert analysis.

References

[1] https://www.w3.org/TR/owl-ref/ [Accessed: 24.10.2025]

[2] Berners-Lee, Tim; James Hendler; Ora Lassila. "The Semantic Web", 2001.

[3] Verhodubs, Olegs. “Prerequisites for Fuzzy Inference on Raw Text Using Semantic Reasoner”, 2025.

[4] Bird, Steven; Edward Loper and Ewan Klein. “Natural Language Processing with Python.” O’Reilly Media Inc, 2009.

[5] Verhodubs, Olegs. “Keyword Search Engine Enriched by Expert System Features”, 2020.

[6] Verhodubs, Olegs. “RUFRON: A library for Generation of Rules from Ontology”, 2024.