

What is the input and output of this program?

The input of this system is the triple you want to define the relationship, and the output is the triple converted to linked data format. In addition, you can use the output value from this method again as an input value to see how it is organically connected with other data. The output value at this time is a diagram of the connection relationship between data.

What is the purpose/intent of this program?

This program was created with two intentions. First, due to the awareness of the problem that the barrier to entry into the semantic web environment is too high, the authoring method using the Venn diagram model preset was introduced so that new users can author linked data without understanding the complex grammar.

Second, in the method of searching the connection relationship between a specific entity and other entities far away in the semantic web environment, the existing approach of searching all connected entities is considered too inefficient, and the depth value between concepts is saved. We devised a method to immediately determine the connection relationship between entities with the same predicate.

Why/How Does This Program Help You?

The two uses/intentions introduced above are helpful for the following reasons. First, since general users can create linked data relationships without difficulty, the tasks that users have to perform multiple times by accessing separate services can be handled at once on the 'web that machines understand themselves'. In particular, in the current Semantic Web, there are often situations in which several services cannot interact due to the absence of a connection relationship. By simply defining a connection relationship, a solution to this problem can be sought.

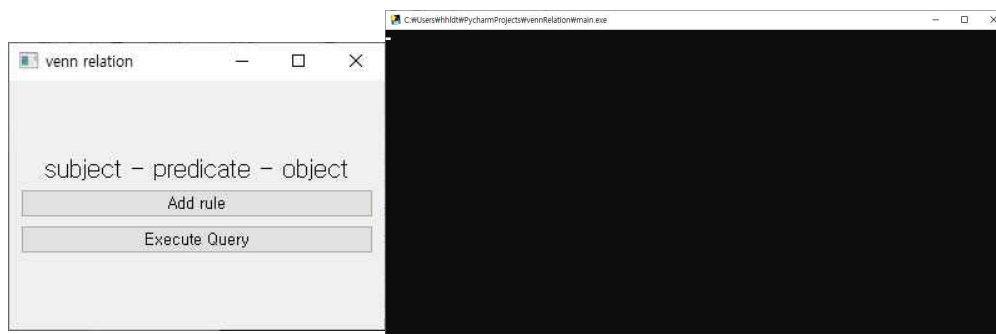
Second, since the inclusion relationship of relations defined in this system stores the inclusion values for each other as predicates according to the Venn diagram model, the hierarchical relationship between entities can be judged immediately. In this system, when using these hierarchical relationships to search for relations according to the level of the hierarchy, we sought a method that is more computationally efficient than the existing method. For example, when selecting an inclusion relationship, the depth index value is given to the selected predicate as the entities a and b are in a parental relationship. The main point is a method of inferring the relationship between a and c that are not connected through index values.

What platform (Windows 10 and what environment) this program will run on

- -This program can be downloaded from

<https://github.com/kellysolow/vennRelation> and it is composed of the exe distribution file, so I think it will work smoothly in most Windows environments. However, because the file size is large, problems may occur when the computer's RAM size is less than 4 GB.

### How to use the system



0) When this system is executed, as shown in the picture above, a ui with buttons for adding rules and executing queries and a console window displaying the output contents appear together.

The console window is used to check the output contents, and there are actually two functions on the left side of the Add rule/query execution.

#### 1) Add rule

The Add Rule button is a function for the user to select two entities to form a relation.

Here, entity means a single entity that can be placed in place of a subject or object. It is the most basic unit, and when the semantic web is a graph composed of numerous nodes and edges, an entity corresponds to a node. Rule refers to the relationship of how two entities are connected to each other, and in the graph example, it acts as an edge. The format of the rule is subject-predicate-object, and these three elements are called triples.

In the semantic web, it is specified how many entities interact with each other through rules and how they relate to each other, and by using these rules, a web (semantic web) that a machine understands itself can be implemented.

However, this program does not address these rules in detail, only the establishment of the most basic linking steps. In other words, if a triple is described in the form of subject -predicate-object, it means 'subject and object are connected by predicate.'

In order to perform the function, the user

1-1) Selecting an entity

1-2) Venn diagram selection

1-3) Input relation in triple form

You will go through the process of.

1-1) entity selection

Users can create a relation by selecting any two words they would like to build a relationship with. As the smallest unit to understand the characteristics of the Semantic Web, you select two words and write what the relationship between them is.

There are no restrictions on words here, but the relation must follow the triple structure of subject-predicate-object.

Given, the entity entering the object can use the existing entity, and the list of existing entities can be checked through the attached entitiesCount.xlsx excel file.

1-2) Venn diagram selection

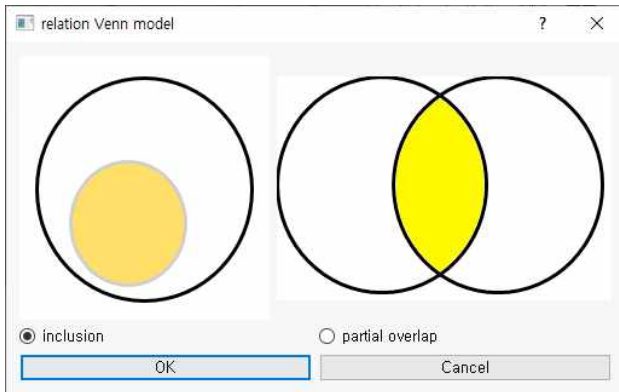
This is the process of making the relation selected in the triple structure in Linked Data format. Those who want to access the existing Semantic Web needed knowledge of the grammar format required for this, but in this program, selecting a Venn diagram can replace this process.

Venn diagram is a mathematical model that visually expresses the relationship between groups, and is not a model that is commonly used in semantic web relations in general. In this study, the Venn diagram was introduced as a simple model that visually expresses the relationship between groups in making relations, and that it is a simple model that general users have encountered frequently.

If you select a relationship similar to the one you want to define among the

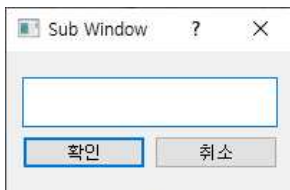
presented Venn diagram models, the relationship defined in the data format of the Semantic Web is automatically converted and saved, and the user does not need to learn about the data format and grammar of the Semantic Web in this process.

ex) As an example of the subject-predicate-object relationship, let's assume the relationship of Korea-subDistrict -Seoul.



Just choose which of the two models fits the relationship you want to define. Here, the model on the left is a relation corresponding to a large circle-subject, and a small colored circle-object. The model on the right is the left circle-subject, the colored part -object, the right circle- corresponds to the object group corresponding to the predicate.

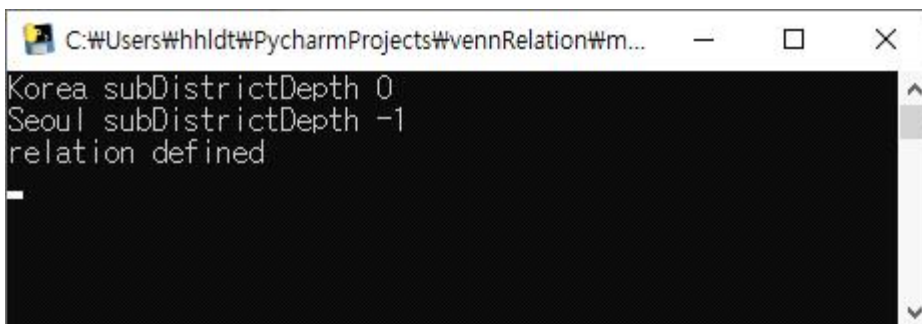
### 1-3) Input relation in triple form



When you select a model and click the OK button, the input window as above appears.

Write the relationship you want to define here as a subject predicate object.

In the example above, you can enter Korea subDistrict Seoul



If the process is successful, you can see the console window as above.

## 2) Query execution

In the relationship corresponding to query execution, you can search for a connection relationship between two entities in a given data dump.

Through this function, the user can search the relationship between the entity established through the rule addition and the relationship between the existing data.

### 2-1) entity selection

The user selects two entities that are curious about the connection relationship.

The entity selected here can be a word created directly in 1) Add rule

It can also be a word that exists in the existing entitiesCount.xlsx file.

Let's search for the existence of the relationship between Korea and Seoul used in the example above.


### 2-2) Enter the query in triple format

Because it searches for what kind of relationship exists, the triple used here is

In the subject-predicate-object format used above, depth is used to determine how precisely to search instead of a predicate.

In other words, it searches in the format of (subject depth object) and scans the given data dump by about  $10^{\text{depth}}$ , so if you reduce the number too high, an error may occur. (5 or less recommended)

The search results for Korea 2 Seoul are shown in the picture below.



```
C:\Users\hhldt\PycharmProjects\vennRelation\m...  
Seoul is connected to Korea through subDistrict  
Seoul is included to Korea  
Korea 0  
Seoul -1  
  
Program finished  
time : 1.1071250438690186
```

In the case of (Korea-Seoul) defined through this system, it is searched much faster because depth is recorded together, and the search for existing entities in entitiesCount.xlsx takes time.

## Scenario

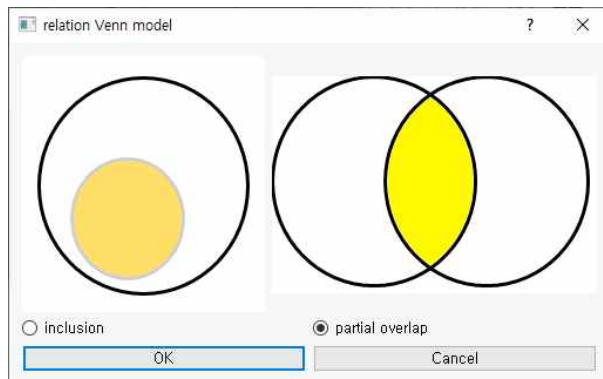
### 1. Michigan-Illinois relationship building

|    |                    |      |
|----|--------------------|------|
| 11 | Michigan           | 1902 |
| 12 | Town               | 1730 |
| 13 | Mountain_Time_Zone | 1599 |
| 14 | Illinois           | 1441 |

While checking the entity that wants to define the relationship in entitiesCount.xlsx, the existence of Michigan and Illinois was confirmed.

As both Michigan and Illinois are states in the United States, this relationship is intended to be defined through a “partial overlap” model that implies that it is an entity of equal depth.

The definition was performed by Michigan StatesOfAmerica Illinois and corresponds to the left circle-Michigan right circle-StatesOfAmerica painted part-Illinois in some overlapping models.



The result is as follows.

```
C:\Users\hhldt\PycharmProjects\ven...  
Michigan StatesOfAmericaDepth 0  
Illinois StatesOfAmericaDepth 0  
relation defined
```

## 2. Seoul - subdistrict\_Korea - Hoegi-Dong Relationship Definition and Search


The triple defined in this system keeps a record for easy future search through the criterion of depth, and 1 depth means the distance to the object connected to one subject.

That is, subject -predicate -object is 1depth

subject - predicate - object (reacts as subject) - predicate - object is 2 depth.

The triple we want to define above is

1. Seoul - subdistrict\_Korea - Dongdaemun-Gu
2. Dongdaemun-Gu - subdistrict\_Korea - Hoegi-Dong




```
C:\Users\whhldt\PycharmProjects\vennRelation\m...
Seoul subDistrict_KoreaDepth 0
Dongdaemun-gu subDistrict_KoreaDepth -1
relation defined
Dongdaemun-gu subDistrict_KoreaDepth -1
Hoegi-dong subDistrict_KoreaDepth -2
relation defined
```

After that, press the query execution button as defined

If you search for,

Seoul 2 Hoegi-dong

you can get the results for Hoegi-dong, Dongdaemun-gu, Seoul.



```
C:\Users\whhldt\PycharmProjects\vennRelation\m...
Hoegi-dong is included to Seoul
Seoul 0
Hoegi-dong -2

Program finished
time : 1.0731031894683838
```

The connection relationship with the value of the existing data dump that does not include the depth requires a calculation time as much as  $10^{\text{depth}}$  to search, but the value defined in the text compares the depth to obtain the result, so it is possible to quickly determine the inclusion relationship. .

### 3. Attaching Big\_Ben to the previously defined England - London

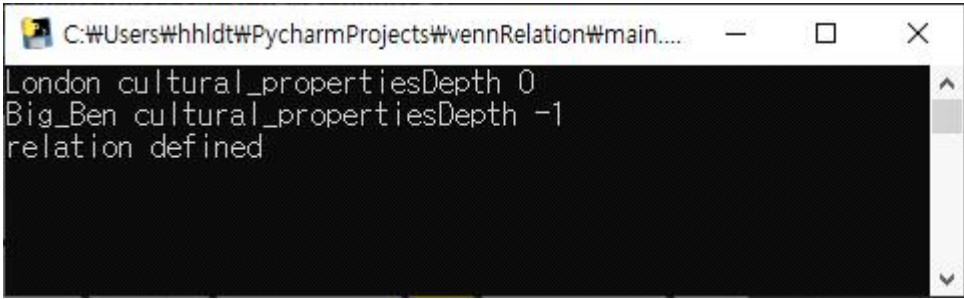
|    |  |     |
|----|--|-----|
| 37 | England  | 597 |
| 38 | Kentucky   | 592 |
| 39 | Alabama  | 572 |
| 40 | Australia  | 566 |
| 41 | France   | 554 |
| 42 | 0.0^^<http://www.w3.org/2001/XMLSchema#float>              | 525 |
| 43 | Alaska   | 518 |
| 44 | English_language   | 467 |
| 45 | 1035995.2441344^^<http://www.w3.org/2001/XMLSchema#double> | 458 |
| 46 | London   | 452 |

Existing dumps contain information about England-capital (predicate)-London.

In this scenario, we want to attach Big\_Ben, the world's most famous clock tower in London.

Add Rule-Select Inclusion Model

London cultural\_properties Big\_Ben



```
C:\Users\hhldt\PycharmProjects\vennRelation\main....  
London cultural_propertiesDepth 0  
Big_Ben cultural_propertiesDepth -1  
relation defined
```

Due to the above definition

England - capital - London - cultural\_properties - Big\_Ben  
relationship was formed.

However, in the case of England, the depth value is not specified in a state that is not directly defined, so unlike scenarios 1 and 2, the search proceeds by searching all objects directly connected to the subject according to the depth.

The results of England 2 Big\_Ben are as follows.



```
depth 1 searching...

using subject : "2011"^^<http://www.w3.org/2001/XMLSchema#gYear>

using subject : Devolution

using subject : Pound_sterling

using subject : London

Found the relation!
```

#### +Depth example London - leader- Boris\_Johnson

Like searching after adding Big\_Ben

It is possible to perform a search according to depth for an entity that already exists without defining it.

data dump<sup>o</sup> London-leader-Boris\_Johnson<sup>o</sup> exists in the data dump

Using this, searching for **England** - capital -London - leader -**Boris\_Johnson** is as follows.

Query execution

England 2 Boris\_Johnson



```
C:\Users\whhldt\PycharmProjects\vonnRelation\main.exe
depth 1 searching...

using subject : "2011"^^<http://www.w3.org/2001/XMLSchema#gYear>

using subject : Devolution

using subject : Pound_sterling

using subject : London

Found the relation!
```