

Prompts used for experiments September 2024

Prompt in natural language (Pattern 1)

You're an ontological engineer. Consider the following two examples:

1. A knowledge base in which an instance of the class "Meeting" is related by a relationship called "Meeting Notes" to an instance of class "Meeting Notes", which is, in turn, related by a relationship called "Contains Action Item" to an instance of class "Action Item". How would you call the direct relationship between the instance of class "Meeting" and the instance of class "Action Item"?
2. A knowledge base in which an instance of the class "Footballer" is related by a relationship called "Got red card in" to an instance of class "Football match", which is, in turn, related by a relationship called "Played at" to an instance of class "Football arena". How would you call the direct relationship between the instance of class "Footballer" and the instance of class "Football arena"?

In both cases, you do not need to use the words from the original names, but rather express the real meaning of the direct relationship that holds between the two classes. And provide 3 possible names for that direct relationship.

Whole conversations

Gemini: <https://gemini.google.com/share/8263beaeb8de>,
<https://gemini.google.com/share/d40d38e2204b>

ChatGPT: <https://chatgpt.com/share/66e3fbfc-dd60-800c-aa38-a4020c41ae74>,
<https://chatgpt.com/share/884fb028-0cb0-4ffa-8094-e9f804e0257f>

Prompt in natural language (Pattern 2)

You're an ontological engineer. Consider a knowledge base in which an instance A of the class "Paper" is related by a relationship called "Has decision" to an instance of class "Acceptance decision". The class "Acceptance decision" is then a subclass of the class "Decision". How would you call a more specific subclass of "Paper" to which instance A belongs? You do not need to use the words from the original names, but rather express the real meaning of the class of papers that have the relationship "Has decision" to an acceptance decision.

Prompt using Turtle syntax (Pattern 1)

1- Zero-shot setting

Task: Consider turtle syntax in the semantic web. I will give you three classes and three object properties, each class and property has a URI and label except the last object property (called `newObjectProperty`) so we put `<blank>` for it's label. We want you to give us 3 suggestions for `<blank>` (just the label). You do not need to use the words from the original names, but rather express the real meaning. put your response in a Python list of strings.

Here is the turtle code:

```
@prefix : <http://www.example.org/test#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#>.
:Organization rdfs:type owl:Class;
  rdfs:label "Organization" .
:ResidenceObject rdfs:type owl:Class;
  rdfs:label "Residence Object" .
:Basicaddress rdfs:type owl:Class;
  rdfs:label "BasicAddress" .
:deliveryLocation a owl:ObjectProperty ;
  rdfs:label "delivery location" ;
  rdfs:domain :Organization ;
  rdfs:range :ResidenceObject .
:address a owl:ObjectProperty ;
  rdfs:label "address " ;
  rdfs:domain :ResidenceObject ;
  rdfs:range :Basicaddress .
:newObjectProperty a owl:ObjectProperty ;
  rdfs:label <blank> ;
  rdfs:domain :Organization ;
  rdfs:range :Basicaddress .
```

GPT-4:

```
Organization      delivery locationResidence Object address  BasicAddress      addres of delivery location
```

```
```python
["primary address", "headquarters location", "main office address"]
```
```

Gold standard: *address of delivery location*

Task: Consider turtle syntax in the semantic web. I will give you three classes and three object properties, each class and property has a URI and label except the last object property (called `newObjectProperty`) so we put `<blank>` for it's label. We want you to give us 3 suggestions for `<blank>` (just the label). You do not need to use the words from the original names, but rather express the real meaning. put your response in a Python list of strings.

Here is the turtle code:

```
@prefix : <http://www.example.org/test#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#>.
:Meeting rdfs:type owl:Class;
  rdfs:label "Meeting" .
:MeetingNotes rdfs:type owl:Class;
  rdfs:label "Meeting Notes" .
:ActionItem rdfs:type owl:Class;
  rdfs:label "Action Item" .
:meetingNotes a owl:ObjectProperty ;
  rdfs:label "Meeting Notes" ;
  rdfs:domain :Meeting ;
  rdfs:range :MeetingNotes .
:containsActionItem a owl:ObjectProperty ;
  rdfs:label "Contains Action Item" ;
  rdfs:domain :MeetingNotes ;
  rdfs:range :ActionItem .
:newObjectProperty a owl:ObjectProperty ;
  rdfs:label <blank> ;
  rdfs:domain :Meeting ;
  rdfs:range :ActionItem .
```

GPT-4:

```
Meeting Meeting Notes      Meeting Notes      Contains Action Item      Action Item      assigned action item
```

```
```python
["Has Action Item", "Includes Task", "Assigns Responsibility"]
```
```



Gold standard: assigned action item

1- Few-shot setting (3)

Task:

Consider turtle syntax in the semantic web. I will give you three classes and three object properties, each class and property has a URI and label except the last object property (called newObjectProperty) so we put <blank> for it's label. We want you to give us 3 suggestions for <blank> (just the label). You do not need to use the words from the original names, but rather express the real meaning. put your response in a Python list of strings.

Here are some example(s) to help you understand our need:

```
@prefix : <http://www.example.org/test#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#>.
:Occurrence rdfs:type owl:Class;
  rdfs:label "Occurrence" .
:Device rdfs:type owl:Class;
  rdfs:label "Device" .
:Interface rdfs:type owl:Class;
  rdfs:label "Interface" .
:isDetectedBy a owl:ObjectProperty ;
  rdfs:label "is detected by" ;
  rdfs:domain :Occurrence ;
  rdfs:range :Device .
:hasInterface a owl:ObjectProperty ;
  rdfs:label "has interface" ;
  rdfs:domain :Device ;
  rdfs:range :Interface .
:newObjectProperty a owl:ObjectProperty ;
  rdfs:label <blank> ;
  rdfs:domain :Occurrence ;
  rdfs:range :Interface .
```

one expected output:["is detected by device with interface"]

```
@prefix : <http://www.example.org/test#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#>.
:Position rdfs:type owl:Class;
  rdfs:label "Position" .
:Company rdfs:type owl:Class;
  rdfs:label "Company" .
:Person rdfs:type owl:Class;
  rdfs:label "Person" .
:company a owl:ObjectProperty ;
  rdfs:label "company" ;
  rdfs:domain :Position ;
  rdfs:range :Company .
:employs a owl:ObjectProperty ;
  rdfs:label "employs" ;
  rdfs:domain :Company ;
  rdfs:range :Person .
:newObjectProperty a owl:ObjectProperty ;
  rdfs:label <blank> ;
  rdfs:domain :Position ;
  rdfs:range :Person .
```

one expected output:["position at company that employs"]

```
@prefix : <http://www.example.org/test#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#>.
:Marketsegment rdfs:type owl:Class;
  rdfs:label "MarketSegment" .
:Marketsellerentity rdfs:type owl:Class;
  rdfs:label "MarketSellerEntity" .
:Marketsegment rdfs:type owl:Class;
  rdfs:label "MarketSegment" .
:seller a owl:ObjectProperty ;
  rdfs:label "seller" ;
  rdfs:domain :Marketsegment ;
  rdfs:range :Marketsellerentity .
:hasMarketSegmentFocus a owl:ObjectProperty ;
  rdfs:label "has Market Segment Focus" ;
  rdfs:domain :Marketsellerentity ;
  rdfs:range :Marketsegment .
:newObjectProperty a owl:ObjectProperty ;
  rdfs:label <blank> ;
  rdfs:domain :Marketsegment ;
  rdfs:range :Marketsegment .
```

one expected output:["seller focused on market segment"]

Here is the turtle code:

```
@prefix : <http://www.example.org/test#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#>.
:Organization rdfs:type owl:Class;
  rdfs:label "Organization" .
:ResidenceObject rdfs:type owl:Class;
  rdfs:label "Residence Object" .
:Basicaddress rdfs:type owl:Class;
  rdfs:label "BasicAddress" .
:deliveryLocation a owl:ObjectProperty ;
  rdfs:label "delivery location" ;
  rdfs:domain :Organization ;
  rdfs:range :ResidenceObject .
:address a owl:ObjectProperty ;
  rdfs:label "address " ;
  rdfs:domain :ResidenceObject ;
  rdfs:range :Basicaddress .
:newObjectProperty a owl:ObjectProperty ;
  rdfs:label <blank> ;
  rdfs:domain :Organization ;
  rdfs:range :Basicaddress .
```

GPT-4:

```
Organization      delivery locationResidence Object address  BasicAddress      addres of delivery location
```

```
``python
["has primary address", "located at address", "registered address"]
``
```

Gold standard: address of delivery location

Task: Consider turtle syntax in the semantic web. I will give you three classes and three three object properties, each class and property has a URI and label except the last object property (called newObjectProperty) so we put <blank> for it's label. We want you to give us 3 suggestions for <blank> (just the label). You do not need to use the words from the original names, but rather express the real meaning. put your response in a Python list of strings.

Here are some example(s) to help you understand our need:

```
@prefix : <http://www.example.org/test#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#>.
:Occurrence rdfs:type owl:Class;
  rdfs:label "Occurrence" .
:Device rdfs:type owl:Class;
  rdfs:label "Device" .
:Interface rdfs:type owl:Class;
  rdfs:label "Interface" .
:isDetectedBy a owl:ObjectProperty ;
  rdfs:label "is detected by" ;
  rdfs:domain :Occurrence ;
  rdfs:range :Device .
:hasInterface a owl:ObjectProperty ;
  rdfs:label "has interface" ;
  rdfs:domain :Device ;
  rdfs:range :Interface .
:newObjectProperty a owl:ObjectProperty ;
  rdfs:label <blank> ;
  rdfs:domain :Occurrence ;
  rdfs:range :Interface .
```

one expected output:["is detected by device with interface"]

```
@prefix : <http://www.example.org/test#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#>.
:Position rdfs:type owl:Class;
  rdfs:label "Position" .
:Company rdfs:type owl:Class;
  rdfs:label "Company" .
:Person rdfs:type owl:Class;
  rdfs:label "Person" .
:company a owl:ObjectProperty ;
  rdfs:label "company" ;
  rdfs:domain :Position ;
  rdfs:range :Company .
:employs a owl:ObjectProperty ;
  rdfs:label "employs" ;
  rdfs:domain :Company ;
  rdfs:range :Person .
:newObjectProperty a owl:ObjectProperty ;
  rdfs:label <blank> ;
  rdfs:domain :Position ;
  rdfs:range :Person .
```

```

one expected output:["position at company that employs"]
@prefix : <http://www.example.org/test#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#>.
:Marketsegment rdf:type owl:Class;
  rdfs:label "MarketSegment" .
:Marketsellerentity rdf:type owl:Class;
  rdfs:label "MarketSellerEntity" .
:Marketsegment rdf:type owl:Class;
  rdfs:label "MarketSegment" .
:seller a owl:ObjectProperty ;
  rdfs:label "seller" ;
  rdfs:domain :Marketsegment ;
  rdfs:range :Marketsellerentity .
:hasMarketSegmentFocus a owl:ObjectProperty ;
  rdfs:label "has Market Segment Focus" ;
  rdfs:domain :Marketsellerentity ;
  rdfs:range :Marketsegment .
:newObjectProperty a owl:ObjectProperty ;
  rdfs:label <blank> ;
  rdfs:domain :Marketsegment ;
  rdfs:range :Marketsegment .

```

one expected output:["seller focused on market segment"]
Here is the turtle code:

```

@prefix : <http://www.example.org/test#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#>.
:Meeting rdf:type owl:Class;
  rdfs:label "Meeting" .
:MeetingNotes rdf:type owl:Class;
  rdfs:label "Meeting Notes" .
:ActionItem rdf:type owl:Class;
  rdfs:label "Action Item" .
:meetingNotes a owl:ObjectProperty ;
  rdfs:label "Meeting Notes" ;
  rdfs:domain :Meeting ;
  rdfs:range :MeetingNotes .
:containsActionItem a owl:ObjectProperty ;
  rdfs:label "Contains Action Item" ;
  rdfs:domain :MeetingNotes ;
  rdfs:range :ActionItem .
:newObjectProperty a owl:ObjectProperty ;
  rdfs:label <blank> ;
  rdfs:domain :Meeting ;
  rdfs:range :ActionItem .

```

GPT-4:

```

Meeting Meeting Notes Meeting Notes Contains Action Item Action Item assigned action item

```python
["has action item", "includes action item", "contains task"]

```

**Gold standard:** assigned action item

## Prompt using Turtle syntax (Pattern 2)

Consider turtle syntax in the semantic web. I will give you four classes and one object property, each class and property has a URI and label except the last class (called News subclass, it is subclass of the first class but we don't have a proper label for it) so we put "???" for its label. We want you to give us 3 suggestions for the label (just the label) that have meaning in the usecase we have (look at code and try to understand and mimic the other subclass name construction). You do not need to use the words from the original names, but rather express the real meaning. put your response in a Python list of strings.

Here is the turtle code:

```
@prefix : http://www.example.org/test# .
@prefix rdfs: http://www.w3.org/2000/01/rdf-schema# .
@prefix owl: http://www.w3.org/2002/07/owl# .

:Paper rdfs:type owl:Class;
rdfs:label "Paper".

:Decision rdfs:type owl:Class;
Rdfs:label "Decision".

:AcceptanceDecision rdfs:type owl:Class;
Rdfs:label "Acceptance Decision";
rdfs:subClassOf :Decision.

:hasDecision rdfs:type owl:ObjectProperty;
rdfs:domain :Paper;
rdfs:range :Decision.

:NewSubClass rdfs:type owl:Class;
rdfs:subClassOf :Paper;
rdfs:label "???" .
```