

1 Updated Scenario

Let's say we know that to parse questions in the target language it is important that the system is capable of dealing with intransitive verbs. However, let's say that the system's library of sentence templates does not contain sentence templates that deal with intransitive verbs.

Thus, we provide a target language expert with

- 1) an example intransitive verb w (e.g., $w = \text{"located (in)"}$) and
- 2) five example questions q , $q - \textit{opposite}$, $q - \textit{amount}$, $q - \textit{opposite} - \textit{amount}$, $q - \textit{ask}$ involving that intransitive verb (e.g., $q = \text{"In which country is Limerick Lake located?"}$, $q - \textit{opposite} = \text{"Which Lake is located in Canada?"}$, $q - \textit{amount} = \text{"In how many countries is the Limerick Lake located?"}$, $q - \textit{opposite} - \textit{amount} = \text{"How many lakes are located in Canada?"}$, and $q - \textit{ask} = \text{"Is Limerick Lake located in Canada?"}$).

Furthermore, we provide the target language expert with

- 3) a set C of syntactic categories (e.g., $C = \{\text{InterrogativeDeterminer, InterrogativePronounForPerson, InterrogativePronounForThing, RelativeNounSingular, RelativeNounPlural, AuxVerb, Determiner, Preposition, PrepositionalAdjunct, ...}\}$) and
- 4) with a dictionary $D_{\textit{simple}}$, that contains a list of words for each syntactic category in C (e.g., $D_{\textit{simple}} = \{ \text{InterrogativeDeterminer} \rightarrow \{ \text{"what"} \}, \text{InterrogativePronounForPerson} \rightarrow \{ \text{"who"} \}, \text{AuxVerb}(\text{be:present:singular}) \rightarrow \{ \text{"is"} \}, \dots \}$).
- 5) Finally, the target language expert receives guidelines about which syntactic categories need to be parameterized and how they need to be parameterized. For example, the syntactic category AuxVerb needs to be parameterized, e.g., which $\text{be:present:singular}$ as $\text{AuxVerb}(\text{be:present:singular})$.
- 6) the guideline also says the target language expert should write sentence templates for five types of given questions: q , $q - \textit{opposite}$, $q - \textit{amount}$, $q - \textit{opposite} - \textit{amount}$, $q - \textit{ask}$

Then, the target language expert is asked to create a sentence template for the given question and for similar questions that involve an intransitive verb. A sentence template is a sequence of syntactic categories from the set C .

For given example q , have a look at the following pair of a question and a sentence template:

q "In which country is Limerick Lake located?"
 ST_q Preposition InterrogativeDeterminer NounSingular AuxVerb(be:present:singular)
Object mainVerb(perfect)?

Besides creating a sentence template for the given question, the target language expert can create further sentence templates, such as:

- InterrogativeDeterminer NounSingular AuxVerb(be:present:singular) Object mainVerb(perfect) Preposition?
(for a question such as "Which country is Limerick Lake located in?")
- InterrogativePlace AuxVerb(be:present:singular) Object mainVerb(perfect)?
(for a question such as "Where is Limerick Lake located?")
- ImperativeVerb(give) pronoun(me) Determiner(the) NounSingular InterrogativePlace Object AuxVerb(be:present:singular) mainVerb(perfect).
(for a question such as "Tell me the country where Limerick Lake is located.")
AuxVerb(do:present:singular) pronoun(you) verb(know) InterrogativePlace
AuxVerb(be:present:singular) Object mainVerb(perfect)?
(for a question such as "Do you know where is Limerick Lake located?")

For given example $q - opposite$,

$q - opposite$ "Which Lake is located in Canada?"

$ST_{q-opposite}$ InterrogativeDeterminer NounSingular AuxVerb(be:present:singular) mainVerb(perfect) Preposition PrepositionalAdjunct?

Besides creating a sentence template for the given question, the target language expert can create further sentence templates, such as:

- InterrogativeDeterminer NounSingular AuxVerb(be:past:singular) mainVerb(perfect) Preposition PrepositionalAdjunct?
(for a question such as "Which Lake was located in Canada?")
- ImperativeVerb(list) Determiner(all) Determiner(the) Subject mainVerb(past) Preposition PrepositionalAdjunct.
(for a question such as "List all the lakes located in Canada.")

For given example $q - amount$,

$q - amount$ "In how many countries is the Limerick Lake located?"

$ST_{q-amount}$ Preposition InterrogativeAmount NounPlural AuxVerb(be:present:singular) Determiner(the) Object mainVerb(perfect)?

Note that, the target language expert did not write additional templates for type $q - amount$.

For given example $q - opposite - amount$,

$q - opposite - amount$ "How many lakes are located in Canada?"

$ST_{q-opposite-amount}$ InterrogativeAmount NounPlural AuxVerb(be:present:plural) mainVerb(perfect) Preposition PrepositionalAdjunct?

Besides creating a sentence template for the given question, the target language expert can create further sentence templates, such as:

- InterrogativeAmount NounPlural AuxVerb(be:past:plural) mainVerb(perfect) Preposition PrepositionalAdjunct?
(for a question such as "How many lakes were located in Canada?")

For given example $q - ask$,

$q - amount$ "Is Limerick Lake located in Canada?"
 $ST_{q-opposite-amount}$ AuxVerb(be:present:singular) Subject mainVerb(perfect) Preposition PrepositionalAdjunct?

Note that, no examples were provided for these four sentence templates. The target language expert wrote these templates by themselves.

2 Grammar Rule Template

from ST to GRT Let's say we got some sentence templates in the target language for intransitive verbs from language experts. How these sentence templates can be transformed into grammar rule templates?

a) This is an automatic process and no manual work is involved in this stage.

Here I am repeating the sentence templates for the question q and additional templates for better understanding:

q "In which country is Limerick Lake located?"
 ST_q Preposition InterrogativeDeterminer NounSingular AuxVerb(be:present:singular) Object mainVerb(perfect)?
– InterrogativeDeterminer NounSingular AuxVerb(be:present:singular) Object mainVerb(perfect) Preposition?
(for a question such as "Which country is Limerick Lake located in?")
– InterrogativePlace AuxVerb(be:present:singular) Object mainVerb(perfect)?
(for a question such as "Where is Limerick Lake located?")
– ImperativeVerb(give) pronoun(me) Determiner(the) NounSingular InterrogativePlace Object AuxVerb(be:present:singular) mainVerb(perfect).
(for a question such as "Tell me the country where Limerick Lake is located.")
AuxVerb(do:present:singular) pronoun(you) verb(know) InterrogativePlace AuxVerb(be:present:singular) Object mainVerb(perfect)?
(for a question such as "Do you know where is Limerick Lake located?")

Transform Sentence template to Grammar rule template in 4 steps: The changes in each step are shown in bold.

Step 1: Add a [] bracket to all the syntactic categories of sentence templates that will be replaced by written form of lexical entries. We call it pre-terminals.
 $D_{pre-terminal} = \{IntransitiveVerb \rightarrow \{Preposition, mainVerb(present), mainVerb(past), mainVerb(perfect)\}\}.$

ST_q [**Preposition**] InterrogativeDeterminer NounSingular AuxVerb(be:present:singular) Object [**mainVerb(perfect)**]?
– InterrogativeDeterminer NounSingular AuxVerb(be:present:singular) Object [**mainVerb(perfect)**] [**Preposition**]?
– InterrogativePlace AuxVerb(be:present:singular) Object [**mainVerb(perfect)**]?

- ImperativeVerb(give) pronoun(me) Determiner(the) NounSingular InterrogativePlace Object AuxVerb(be:present:singular) [**mainVerb(perfect)**].
- AuxVerb(do:present:singular) pronoun(you) verb(know) InterrogativePlace AuxVerb(be:present:singular) Object [**mainVerb(perfect)**]?

Step 2: replace all the syntactic categories of sentence templates from the following dictionary: $D_{non-terminal} = \{ \text{PrepositionalAdjunct} \rightarrow \langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle, \text{Object} \rightarrow \langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle, \text{NounSingular} \rightarrow \langle \text{NP}_{\text{Class}}, \langle \text{Property}, \text{Range} \rangle \rangle, \text{NounPlural} \rightarrow \langle \text{NP}_{\text{Class}}, \langle \text{Property}, \text{Range} \rangle \rangle \}$

- ST_q [Preposition] InterrogativeDeterminer $\langle \text{NP}_{\text{Class}}, \langle \text{Property}, \text{Range} \rangle \rangle$ AuxVerb(be:present:singular) $\langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle$ [mainVerb(perfect)]?
1. InterrogativeDeterminer $\langle \text{NP}_{\text{Class}}, \langle \text{Property}, \text{Range} \rangle \rangle$ AuxVerb(be:present:singular) $\langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle$ [mainVerb(perfect)] [Preposition]?
 2. InterrogativePlace AuxVerb(be:present:singular) $\langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle$ [mainVerb(perfect)]?
 3. ImperativeVerb(give) pronoun(me) Determiner(the) $\langle \text{NP}_{\text{Class}}, \langle \text{Property}, \text{Range} \rangle \rangle$ InterrogativePlace AuxVerb(be:present:singular) $\langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle$ [mainVerb(perfect)].
 4. AuxVerb(do:present:singular) verb(know) InterrogativePlace AuxVerb(be:present:singular) $\langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle$ [mainVerb(perfect)]?

Step 3: replace all the syntactic categories (with or without parameters) using the dictionary $D_{simple} = \{ \text{InterrogativeDeterminer} \rightarrow \{ \text{"what"} \}, \text{InterrogativePronounForPerson} \rightarrow \{ \text{"who"} \}, \text{AuxVerb}(\text{be:present:singular}) \rightarrow \{ \text{"is"} \}, \text{ImperativeVerb}(\text{present}) \rightarrow \{ \text{"give"} \}, \dots \}$.

After Step 3, the sentence template ST_q becomes Grammar rule template GRT_q :

- GRT_q [Preposition] **which** $\langle \text{NP}_{\text{Class}}, \langle \text{Property}, \text{Range} \rangle \rangle$ **is** $\langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle$ [mainVerb(perfect)]?
1. **Which** $\langle \text{NP}_{\text{Class}}, \langle \text{Property}, \text{Range} \rangle \rangle$ **is** $\langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle$ [mainVerb(perfect)] [Preposition]?
 2. **Where is** $\langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle$ [mainVerb(perfect)]?
 3. **Give me the** $\langle \text{NP}_{\text{Class}}, \langle \text{Property}, \text{Range} \rangle \rangle$ **where** $\langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle$ **is** [mainVerb(perfect)].
 4. **Do you know where is** $\langle \text{NP}_{\text{map}(\text{SyntacticFunction}), \text{Property}} \rangle$ [mainVerb(perfect)]?

Step 4:

Similarly, do Step 1, Step 2, and Step 3 for all other sentence templates of $ST_{q-opposite}$, $ST_{q-amount}$, $ST_{q-opposite-amount}$, and ST_{q-ask}

For example, after Step 1, Step 2, and Step 3 the sentence templates $ST_{q-opposite}$ it becomes Grammar rule template $GRT_{q-opposite}$.

$GRT_{q-opposite}$ Which $\langle NP_{Class}, \langle Property, Range \rangle \rangle$ is [mainVerb(perfect)] [Preposition]
 $\langle NP_{map(SyntacticFunction)}, Property \rangle$?
1. Which $\langle NP_{Class}, \langle Property, Range \rangle \rangle$ were [mainVerb(perfect)] [Preposition]
 $\langle NP_{map(SyntacticFunction)}, Property \rangle$?
2. List all the $\langle NP_{Class}, \langle Property, Range \rangle \rangle$ [mainVerb(past)] [Preposition]
 $\langle NP_{map(SyntacticFunction)}, Property \rangle$.

For example, after Step 1, Step 2, and Step 3 for sentence templates $ST_{q-amount}$ it becomes Grammar rule template $GRT_{q-amount}$.

$GRT_{q-amount}$ [Preposition] how many $\langle NP_{Class}, \langle Property, Range \rangle \rangle$ is the $\langle NP_{map(SyntacticFunction)}, Property \rangle$
[mainVerb(perfect)]?

For example, after Step 1, Step 2, and Step 3 for sentence templates $ST_{q-opposite-amount}$ it becomes Grammar rule template $GRT_{q-opposite-amount}$.

$GRT_{q-opposite-amount}$ How many $\langle NP_{Class}, \langle Property, Range \rangle \rangle$ are [mainVerb(perfect)] [Preposition]
 $\langle NP_{map(SyntacticFunction)}, Property \rangle$?
1. How many $\langle NP_{Class}, \langle Property, Range \rangle \rangle$ were [mainVerb(perfect)] [Preposition]
[mainVerb(perfect)] $\langle NP_{map(SyntacticFunction)}, Property \rangle$?

For example, after Step 1, Step 2, and Step 3 for sentence templates ST_{q-ask} it becomes Grammar rule template GRT_{q-ask} .

For given example $q - ask$,

$q - amount$ "Is Limerick Lake located in Canada?"

$ST_{q-opposite-amount}$ Is $\langle NP_{map(SyntacticFunction)}, Property \rangle$ [mainVerb(perfect)] [Preposition] $\langle NP_{map(SyntacticFunction)}, Property \rangle$?

Therefore, we achieved the grammar rule templates : GRT_q , $GRT_{q-opposite}$, $GRT_{q-amount}$, $GRT_{q-opposite-amount}$, and GRT_{q-ask}

The $\langle NP_{map(SyntacticFunction)}, Property \rangle$ of (grammar rule templates: GRT_q , $GRT_{q-opposite}$, $GRT_{q-amount}$, $GRT_{q-opposite-amount}$, and GRT_{q-ask}) is instantiated either to $\langle NP_{Domain}, Property \rangle$ or $\langle NP_{Range}, Property \rangle$, depending on whether mapSyntacticFunction is Domain or Range. $\langle NP_{map(SyntacticFunction)}, Property \rangle$ the corresponding labels of URIs using the following SPARQL queries:

```
<NPDomain, Property> -> eval(SELECT ?label WHERE {?Domain Property ?Range .  
?Domain rdfs:label ?label . })  
<NPRange, Property> -> eval(SELECT ?label WHERE {?Domain Property ?Range .
```

```

                                ?Range rdfs:label ?label . })
<NPClass, <Property,Domain>> -> eval(SELECT ?label WHERE {?Domain Property ?Range.
                                ?Domain rdf:type ?Class.
                                ?Class rdfs:label ?label })
<NPClass, <Property,Range>> -> eval(SELECT ?label WHERE {?Domain Property ?Range.
                                ?Range rdf:type ?Class.
                                ?Class rdfs:label ?label })

```

all the sections below, we are re-written above. Therefore, all the sections and discussions are the previous discussion.

3 Given Scenario

a) Can we build on the scenario that there is a question that cannot be parsed yet, so the goal is first to create a sentence template? -In the current system, we can not parse the following question: *q*: "In which country is Limerick Lake located?".

b) What pieces of information do we need to provide the language expert with?

- The language expert is given the question *q* - The language expert is given a dictionary D_{simple} .

$D_{simple}:C = \{\text{InterrogativeDeterminer, InterrogativePronounForPerson, InterrogativePronounForThing, RelativeNounSingular, RelativeNounPlural, AuxVerb(be:present:singular), Determiner, Preposition, PrepositionalAdjunct, ...}\}.$

InterrogativeDeterminer » "what"
InterrogativePronounForPerson » "who"
AuxVerb(be:present:singular) » "is"
AuxVerb(be:past:singular) » "was"
AuxVerb(be:present:plural) » "are"
AuxVerb(be:past:plural) » "were"
Determiner » "the",
mainVerb(present)
mainVerb(present 3rd person)
mainVerb(past)
mainVerb(past) etc.

c) What does a language expert need to do to create the sentence template? The language expert takes the question *q* as an example and writes a sentence template

Preposition InterrogativeDeterminer Noun AuxVerb(be:present:singular) Object Determiner mainVerb(perfect) (In which country is Limerick Lake located?)

Here I answered as per the scenario you provided. Note that, in your provided scenario, we already have an English System and we cannot parse a question

and we want the question to be parsed. We tell the language expert to write a sentence template for the question. This is not the goal of writing a sentence template by a language expert. For one question and English, it can be done by replacing tokens from D_{simple} .

Sometimes we can not provide an example. For the Bangla (or Arabic) QA system, we don't have an example. In that case, we only ask the language experts to write sentence templates for intransitive verbs and give an English example q so that the expert can understand what we want. Note that, replacing tokens D_{simple} does not work here, since we don't have a Bangla example.

4 the Scenario supposed to be

a) So, the scenario is supposed to be: we want to parse sentences in English that contain an intransitive verb and a preposition. That is We asked an English language expert to write templates for that. We provide an example q : "In which country is Limerick Lake located?" so that the language expert can understand what we mean. The question is just an example. So language expert writes the following templates:

b) What pieces of information do we need to provide the language expert with?

- The language expert is given the question q - The language expert is given a dictionary D_{simple} - The language expert is instructed to write sentence templates for intransitive verb.

c) Now the language expert can write:

ST-1: Preposition InterrogativeDeterminer Noun AuxVerb(be:present:singular) Object mainVerb(perfect) (1. In which country is Limerick Lake located?)

ST-2: InterrogativeDeterminer Noun AuxVerb(be:present:singular) Object Object mainVerb(perfect) Preposition (2. which country is Limerick Lake located in?)

ST-3: InterrogativePlace AuxVerb(be:present:singular) Object (3. Where is Limerick Lake located?)

ST-4: ImperativeVerb(present) pronoun(me) Determiner Noun Interrogative-Place Object AuxVerb(be:present:singular) mainVerb(perfect) (4. Tell me the country where Limerick Lake is located.)

ST-5:AuxVerb(do) pronoun(you) verb(know) InterrogativePlace AuxVerb(be:present:singular) Object mainVerb(perfect) (5. Do you know where is Limerick Lake located?)

Note that, no examples were provided for ST-2 to ST-5. The language expert wrote these templates by himself.

5 Writing Sentence Template

A language expert is given a question (i.e., an example question) q and a list C of syntactic categories. In his language, he is told to write sentence templates(STs) for relative nouns (i.e., NounPPFrame in our case). Our goal is to get different variations of sentences containing relative nouns.

Question 1: the language expert would not get any question. If our goal is to get variations for relative nouns, then one needs to use a question that contains a relative noun, right? But, maybe we need to simplify the discussion. Instead of having the goal to learn linguistic variations, we just want to have a sentence template for a question our system cannot process yet?

Example:

$q = \text{"Who is the founder of Paris?"}$

$C = \{\text{InterrogativePronounForPerson, InterrogativePronounForThing, RelativeNounSingular, RelativeNounPlural, AuxVerb, Determiner, Preposition, PrepositionalAdjunct, ...}\}$ and the dictionary D_{simple} . D_{simple} is dictionary for tokens of syntactic category (C): **InterrogativePronounForPerson** $\rightarrow \{\text{"who"}\}$, **Determiner** $\rightarrow \{\text{"the"}\}$, etc.

*Question 2: the dictionary maps a category to a set of words, right? Not necessarily only to a single word? Thus, I wrote **Determiner** $\rightarrow \{\text{"the"}\}$ instead of **Determiner** = "the".*

Question 3: I guess there can be words that can be in multiple categories. In these cases, the language expert needs to decide on the category? Otherwise, we could automatically replace tokens by categories, right?

The language expert writes ~~transforms the question q into a sentence template (ST) by replacing tokens in the question~~ by categories from C ~~in two steps. the question q is just an example of relative noun and it is given to help language experts. Language experts often write more variations of sentence templates than the q .~~

Note that, the q is just an example that helps the language expert to write sentence templates. Say, When we worked on Bangla language, The language expert of Bangla was instructed to write Bangla sentence templates for relative nouns and the English question 'who is the founder of Paris' is given as an example just to help them to think of Bangla templates for the relative noun.

Comment: this is again about how we can obtain linguistic variations. Let's just focus on how sentence templates are created, because at the moment I'd like to understand how sentence templates are created and used. Also, let's not make it complicated by talking about different languages.

First, the language expert needs to decide which frame f is used in the question. Here, the language expert needs to select an element from the set $F = \{ \text{NounPPFrame}, \dots \}$. Here, the language expert correctly decides to select $f = \text{NounPPFrame}$, because in the question there is a singular noun (i.e., "founder") followed by a preposition (i.e., "of") followed by a prepositional adjunct (i.e., "Paris"). As I mentioned at the beginning the language expert is instructed that he has to write templates for relative nouns and an example q is given. so he knows it is a relative noun (which is in our case NounPPFrame). The term 'relative noun' is easy for him to understand. He does not need to find out it is a NounPPFrame .

In the case of English, for the question q , the language expert wrote $q1'$ and also wrote more templates of relative noun (i.e., $q2'$) one token in the question is replaced by **RelativeNounSingular**, another token is replaced by **Preposition**, and another token is replaced by **PrepositionalAdjunct**. "who" is replaced by **InterrogativePronounForPerson**, "is" replaced by **AuxVerb(be:present:singular)**, "the" is replaced by **Determiner**. The language expert uses a set of guidelines that tells them how to select frames.

Thus, the intermediate result is:

The sentence templates written by language expert :

$q' = \text{"Who is the [RelativeNounSingular] [Preposition] [PrepositionalAdjunct] ?"}$

$q1' = \text{"InterrogativePronounForPerson AuxVerb(be:present:singular) Determiner RelativeNounSingular Preposition PrepositionalAdjunct ?"}$

$q2' = \text{"InterrogativePronounForPerson AuxVerb(be:past:singular) Determiner RelativeNounSingular Preposition PrepositionalAdjunct ?"}$

$f = \text{NounPPFrame}$ They know it is construction of relative noun which is NounPPFrame in the system.

In the second step, the other tokens are also replaced by elements of C .

I think we can remove the second step. it is not necessary conceptually

Example:

$q'' = \text{"}e_0 e_1 e_2 [\text{NounSingular}] [\text{Preposition}] [\text{PrepositionalAdjunct}] e_3 \text{"}$

$e_0 \dots e_3$ are elements of C . For our explanation it does not matter which categories exactly.

Question: do we conceptually need two steps?

no, we can remove it.

Question: Can we automatically replace the other tokens in the sentence, by using a dictionary D_{simple} , where that dictionary for example specifies that "Who" belongs to the category **InterrogativePronounForPerson**? Also, can we express that dictionary as a list of sets, such as $\text{"InterrogativePronoun"} = \{\text{Who}\}$?

We do have resources for each language for replacing. And we replace all these c_0 from that. Here is the resource of english: i.e., D_{simple} <https://github.com/fazleh2010/multilingual-grammar-generator/blob/journal/src/main/resources/en/base/base.ttl>

Question 4: wait, if we can automatically replace the tokens in the question by categories, what does the language expert actually do?

We call q a generated rule template (GRT). Or now I am confused. Isn't that the sentence template?

yes, $q1'$ and $q2'$ is sentence templates not grammar rule templates(GRTs). The next step from sentence template is Grammar rule templates.

The sentence template(STs) is an abstracted version of the original question which we can now concretize so that many different questions can be processed. The sentence template is written by a language expert and it does not say which part is pre-terminal and which part is non-terminal. It is purely a syntactic structure of the sentence. The question given is just an example that helps the language expert to write templates. The goal is to get syntactic variations of relative nouns for English.

6 Sentence Template(ST) to Grammar Rule Template(GRT)

Now from these sentence templates(STs), the Grammar rule templates (GRTs) are defined/generated by using D_{simple} and L_{Lemon} . D_{simple} is dictionary for tokens of syntactic category: **InterrogativePronounForPerson**»"who", **Determiner**»"the", etc. L_{Lemon} is lemon lexical contains lexical entries created by language expert.

- We look into lexical entries of L_{Lemon} for **NounPPFrame**, for example '*mayor*' (*of*), '*boardgame*' (*by*), etc. It says that a syntactic structure of copulative construction, in which **Prepositional Adjunct** is either the subject of the property (for the case of '*mayor*' (*of*), (domain|dbo:mayor) or object of the property (for the case of '*boardgame*' (*by*), (range|dbo:publisher). So, **Prepositional Adjunct** of $q1'$ and $q2'$ are replaced by a non-terminal ($\langle \rangle$) using a generic function (i.e., $\langle NP_{map(SyntacticFunction), Property} \rangle$). The $\langle NP_{map(SyntacticFunction), Property} \rangle$ is either to $\langle NP_{Domain, Property} \rangle$ (for example for "mayor" of) or $\langle NP_{Range, Property} \rangle$ (for example for "boardgame" by), depending on whether mapSyntacticFunction is **Domain** or **Range**.
- The lexical entries of **NounPPframe** of L_{Lemon} contains written from of **RelativeNounSingular** and **Preposition**. This information can be achieved from lexical entries and they are pre-terminals(\square). **RelativeNounSingular** and **Preposition** of $q1'$ and $q2'$ are replaced by replaced by [NounSingular] and [Preposition]
- D_{simple} contains **InterrogativePronounForPerson** is "who", **AuxVerb**(be:present:singular) is "is", **Determiner** is "the"

So Form sentence templates $q1'$ and $q1'$, the Grammar rule templates are created using D_{simple} and L_{Lemon} :

$GRT'_1 = \text{"Who is the [NounSingular] [Preposition] <NP}_{map(SyntacticFunction), Property}>?"$

$GRT'_2 = \text{"Who was the[NounSingular] [Preposition] <NP}_{map(SyntacticFunction), Property}>?"$

The sparql query for the non-terminal $<NP_{map(SyntacticFunction), Property}>$ is $<NP_{Domain, Property}>$
 $\rightarrow \text{eval(SELECT ?label WHERE \{?Domain Property ?Range . ?Domain rdfs:label ?label . \}) <NP}_{Range, Property}> \rightarrow \text{eval(SELECT ?label WHERE \{?Domain Property ?Range . ?Range rdfs:label ?label . \})}$

7 Grammar Rule Template(GRT) to Grammar Rule(GR)

The grammar rules are created by simply instantiating grammar rule templates discussed in the previous section.. We start with lemon lexica and iterate over all lexical entries of **NounPpFrame**.

For example for lexical entry "*mayor*" (*of*), the GRT'_1 and GRT'_2 are transform to grammar rules $GR1'$ and $GR2'$

$GR1' = \text{"Who is the mayor of <NP}_{Domain, Property}>?"$

$GR2' = \text{"Who was the mayor of <NP}_{Domain, Property}>?"$

The sparql query for non-terminal $<NP_{Domain, Property}>$ is $<NP_{Domain, dbo:mayor}>$
 $\rightarrow \text{eval(SELECT ?label WHERE \{?Domain dbo:mayor ?object. ?Domain rdfs:label ?label . \})}$ The sparql query for the grammar rule is $\text{SELECT ?label WHERE \{?Domain dbo:mayor ?Range. ?Range rdfs:label ?label . \})}$

8 Grammar Rule(GR) as regular expression

After that, the non-terminals of the grammar rules are then transformed into regular expressions: $GR1' = \text{"Who is the mayor of (.*)?"}$

$GR2' = \text{"Who was the mayor of (.*)?"}$

The sparql query for non-terminal $<NP_{Domain, Property}>$ is $(.*)? \rightarrow \text{eval(SELECT ?label WHERE \{?Domain dbo:mayor ?object. ?Domain rdfs:label ?label . \})}$

The sparql query for the grammar rule is $\text{SELECT ?label WHERE \{?Domain dbo:mayor ?Range. ?Range rdfs:label ?label . \})}$

~~Therefore, we make use of another dictionary, $D_{KG=ret}$ that has entries that are more complex than those in D_{simple} .~~

~~For example, it can contain an entry that expresses that "mayor (of)" is a NounPPFrame, the KG property is "dbo:mayor", and that the value of the domain-or-range property is "domain".~~

~~Given this entry, we can concretize the GRT into a rule template (RT). We can use that entity because we know that the frame in the question in NounPPFrame and the type of the lexical entry is NounPPFrame.~~

~~=~~

~~Example:~~

~~"Who is the mayor of <domain|dbo:mayor>?" (a rule template)~~

~~Here, "domain" is the value of the property domain-or-range of the lexical entry,~~

and "dbo:mayor" is the value of the property property of the lexical entry. The other categories, c0...c3, have been replaced using entries in D_{simple} . Note that " $\langle domain|dbo:mayor \rangle$ " is a non-terminal symbol — the other tokens are terminal symbols.

This all happens offline. These grammar rules are used for parsing a question to SPARQL Query.

9 Parsing

Now, online, the QA system receives a user-provided question such as "Who is the mayor of London?". The system needs to decide which rule template grammar rule can be applied. First, the grammar rules template are transformed into a regular expression, e.g., "Who is the mayor of (.*)?"

This regular expression $GR1'$ matches the question. We store the result of the matching group in the variable $\$Z$ (here, $Z = \text{"London"}$). $GR1' = \text{"Who is the mayor of (.*)?"}$

The sparql query for non-terminal $\langle NP_{Domain, Property} \rangle$ is $(.*) \rightarrow \text{eval}(\text{SELECT ?label WHERE \{?Domain dbo:mayor ?object. ?Domain rdfs:label ?label . \}})$
The sparql query for the grammar rule is $\text{SELECT ?label WHERE \{?Domain dbo:mayor ?Range. ?Range rdfs:label ?label . \}}$

Then we run the SPARQL query $\$Z \rightarrow \text{eval}(\text{SELECT ?label WHERE \{?Domain dbo:mayor ?object. ?Domain rdfs:label ?label . \}})$

This list contains tuples such as $\{(dbr:Paris, "Paris"), (dbr:Berlin, "Berlin"), (dbr:London, "London"), \dots\}$. We find the entity that has a label which is most similar to the value bound to the variable $\$Z$. (The similarity function is based on Jaccard Similarity.).

. Therefore we found $\$Z \rightarrow \text{London}$

So the SPARQL query of the question as follows: and it is parse result of the question The sparql query for the grammar rule is $\text{SELECT ?label WHERE \{dbr:London dbo:mayor ?Range. ?Range rdfs:label ?label . \}}$

Then, we query DBpedia to obtain the list of entities that can be subject of triples with the predicate `dbo:mayor`, together with their labels. The SPARQL query is generated automatically. Here: $\text{SELECT ?x ?z WHERE \{ ?x dbo:mayor ?y. ?y rdfs:label ?z.\}}$.

Then, we generate a SPARQL query that answers the question: $\text{SELECT ?res WHERE \{ \$Z dbo:mayor ?res \}}$. (We would generate " $?res \text{ dbo:mayor } \Z " if the lexical entry would specify `domain-or-range=range`.)

Thus, we have an approach that can parse a question such as "Who is the mayor of London?" into the corresponding SPARQL query and by evaluating the SPARQL query we answer the question using a knowledge base. Note that initially we began with a question about the founder of Paris [Sentence template\(ST\)](#) for relative noun given an example question about the founder of

Paris, The language experts wrote many variations of relative nouns. These sentence templates(STs) are converted to grammar rule templates(GRTs) using D_{simple} and L_{Lemon} . after we instantiate the grammar rule templates(GRTS) by all lexical entries and then creates grammar rules. The grammar rules are then used for parsing a question "who is the mayor of London". The system can also parse "who is the mayor of London".

~~generalizing and concretizing we can now also process/parse a question about the mayor of London.~~