Antlr Grammar Rule Syntax

# Revisions

|  |  |  |
| --- | --- | --- |
| **Date** | **Name** | **Comment** |
| 5/5/2014 | Cheng Quan | Draft document of NLQ grammar |
| 8/22/2014 | Cheng Quan | Update NLQ grammar |

Contents

[Revisions 1](#_Toc396491676)

[Introduction 2](#_Toc396491677)

[Basic knowledge of grammar rules 2](#_Toc396491678)

[Grammar rule syntax 2](#_Toc396491679)

[Grammar notation 3](#_Toc396491680)

[Semantic predicate 3](#_Toc396491681)

[Syntactic predicate 3](#_Toc396491682)

[Description of NLQ grammar rules 4](#_Toc396491683)

[Root rule 4](#_Toc396491684)

[Sub-rules 4](#_Toc396491685)

[Leaf rules 13](#_Toc396491686)

[Antlr Bugs 14](#_Toc396491687)

# Introduction

Antlr grammar is essentially a grammar declaration followed by a list of rules. Each rule can be composed of some sub-rules. Antlr will parse the input sentence from the root rule to the leaf rule and generate a parse tree in the process of parsing. Figure 1 shows the parse tree of input sentence “give me the top 3 call center by sales”.

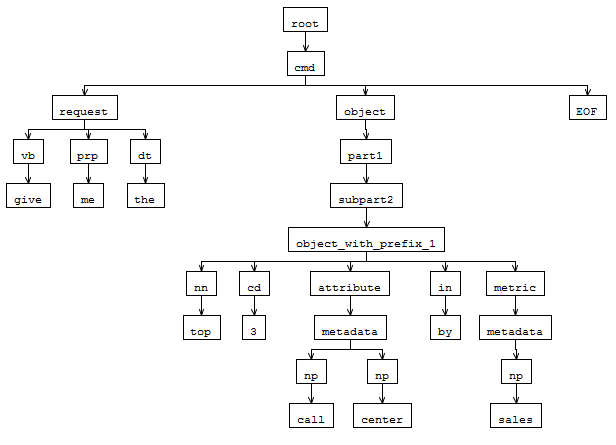


Figure 1

# Basic knowledge of grammar rules

## **Grammar rule syntax**

**Type Syntax**

Non-leaf rule Rule name: {Semantic predicate}? Sub-rules;

Leaf rule Rule name: {Semantic predicate}? Tokens;

Note: Semantic predicate is optional.

## **Grammar notation**

**Syntax Description**

**x Match token or sub-rule x.**

**x y ... z Match a sequence of rule elements.**

**(... | ... | ...) Sub-rule with multiple alternatives.**

**x? Match x or skip it.**

**x\* Match x zero or more times.**

**x+ Match x one or more times.**

## **Semantic predicate**

**Semantic predicates, {...}?, are Boolean expressions written in the target language that indicate the validity of continuing the parse along the path guarded by the predicate. That means Antlr will ignore the sub-rules or tokens guarded with predicates that currently evaluate to false.**

**Below is an example of semantic predicate:**

**metric: {$cmd::mpMDSearch->CheckMetric()}? metadata;**

**This rule means that the metadata will be identified as metric only if the result of function CheckMetric is true.**

## **Syntactic predicate**

Syntactic predicates, (…)=>, specify the syntactic validity of applying an alternative and indicate syntactic context that must be satisfied in order for an alternative to match. Syntactic predicate is often used when multiple alternatives have the same prefix.

Below is an example of syntactic predicate:

expr: (in metric)=>in metric | (in attribute)=>in attribute;

This rule means: if the sub-rule sequence is “in metric”, rule expr will match the 1st alternative “in metric”; if the sub-rule sequence is “in attribute”, rule expr will match the 2nd alternative “in attribute”.

Without syntactic predicate, the rule expr is below:

expr: in metric | in attribute;

In this case, rule expr will never match the 2nd alternative as these two alternatives have the same prefix “in”. When Antlr match the prefix “in” of the 1st alternative, it will go to the branch of 1st alternative and never backtrack to the 2nd alternative.

# ****Description of NLQ grammar rules****

## **Root rule**

**Syntax**

**cmd:** [request](#request) **(**[exist\_report](#exist_report) **| (cc?** [object](#object)**)+) punct? EOF;**

**Description**

This rule will match the total input sentence. The input sentence can be a request of an existing report or a report with multi-objects. An object can be an existing filter, an attribute filter, a metric filter, etc. The input sentence can be ended with a punctuation.

**Examples**

Give me the top 10 call centers in sales last month

Give me the call centers in sales greater than 1000 in the year of 2014

## **Sub-rules**

**Syntax**

request: (wh (vb | pos) | md? prp? [imperative](#imperative)? prp?) dt?;

**Description**

This rule will match the request part of the imperative input sentence.

**Examples**

Where is

What’s

Give me the

Can you show me the

**Syntax**

exist\_report: {$cmd::mpMDSearch->CheckReport()}? [exist\_report\_name](#exist_report_name);

exist\_report\_name: TERM+;

**Description**

These rules will match existing report. A phrase will be identified as existing report only if the result of function CheckReport is true.

**Examples**

top 10 call centers in sales last month

**Syntax**

object: [part1](#part1) | [part2](#part2);

part1: [subpart1](#subpart1) | [subpart2](#subpart2);

part2: [subpart3](#subpart3) | [subpart4](#subpart4);

**Description**

As rule object has too many sub-rules, these sub-rules are divided into two groups. For the same reason, rule part1 and part2 are divided into two groups.

**Syntax**

subpart1: [exist\_filter](#exist_filter) | [attribute\_filter](#attribute_filter);

**Description**

This rule will match existing filter or attribute filter.

**Examples**

my call centers

NY and DC

**Syntax**

subpart2: [object\_with\_prefix](#object_with_prefix) | [further\_operation](#further_operation);

**Description**

This rule will match the object with prefix like “top 3” and “best”, which will lead to a filter on derived metric or further operation words.

**Examples**

top 3 call centers in sales

drill down

**Syntax**

subpart3: [metric\_filter](#metric_filter) | graph\_type;

**Description**

This rule will match metric filter or graph type.

**Examples**

revenue greater than 1,000,000 dollars

pie chart

**Syntax**

subpart4: metadata | connection;

**Description**

This rule will match raw metadata or connection words.

**Examples**

call center

by the

**Syntax**

object\_with\_prefix: prps? ([ascend](#ascend) | [descend](#descend)) cd? [percent](#percent)? [attribute](#attribute)? in? dt? ((exist\_filter)=>exist\_filter | (attribute in? element)=>inside\_attribute\_filter | (element)=>attribute\_filter)? in? metric;

**Description**

This rule will match the object with prefix like “top 3” which will lead to a filter on derived metric. What’s more, there can be an existing filter or attribute filter inside this rule.

**Examples**

my top 25 percent call centers in NY and DC by sales

top 3 call centers in my region by revenue

top 3 call centers in the category of movies by sales

my worst call center in NY and DC by sales

best call centers in my region by revenue

best call centers in the category of movies by sales

**Syntax**

metadata: ([attribute](#attribute) | [metric](#metric));

**Description**

This rule will match the raw metadata, attribute or metric.

**Examples**

call center

unit sold

**Syntax**

connection: (in | vb | to) dt?;

**Description**

This rule will match the further operation after the first query.

**Examples**

by the

**Syntax**

further\_operation: vb prp? rp?;

**Description**

This rule will match the further operation after the first query.

**Examples**

drill down

**Syntax**

inside\_attribute\_filter: [attribute](#attribute) in? (cc? [element](#element))+;

**Description**

This rule will match the attribute filter inside another rule.

**Examples**

category of books and movies

**Syntax**

attribute\_filter: [element](#element) (cc? [element](#element))\*;

**Description**

This rule will match the attribute filter.

**Examples**

NY and DC

**Syntax**

metric\_filter: [metric](#metric) op in? cd np?;

**Description**

This rule will match metric filter.

**Examples**

revenue greater than 1,000,000 dollars

**Syntax**

metric: [metric\_part\_2](#metric_part_2) | [metric\_part\_1](#metric_part_1);

metric\_part\_1: [metric2](#metric2) | [metric1](#metric1);

metric\_part\_2: [metric4](#metric4) | [metric3](#metric3);

**Description**

These rules will match metric. As rule metric has too many sub-rules, these sub-rules are divided into two groups. For the same reason, rule metric\_part\_1 and metric\_part\_2 are divided into two groups.

**Syntax**

metric1: {$cmd::mpMDSearch->CheckMetric()}? np

**Description**

This rule will match metric with one word. A word will be identified as metric1 only if the result of function CheckMetric is true.

**Examples**

revenue, sales

**Syntax**

metric2: {$cmd::mpMDSearch->CheckMetric()}? np (vb | np)

**Description**

This rule will match metric with two words. A phrase will be identified as metric2 only if the result of function CheckMetric is true.

**Examples**

unit sold

**Syntax**

metric3: {$cmd::mpMDSearch->CheckMetric()}? np in np

**Description**

This rule will match metric with three words. A phrase will be identified as metric3 only if the result of function CheckMetric is true.

**Syntax**

metric4: {$cmd::mpMDSearch->CheckMetric()}? (np | jj) np (np | in) np

**Description**

This rule will match metric with four words. A phrase will be identified as metric4 only if the result of function CheckMetric is true.

**Syntax**

attribute: [attribute\_part\_2](#attribute_part_2) | [attribute\_part\_1](#attribute_part_1);

attribute\_part\_1: [attribute2](#attribute2) | [attribute1](#attribute1);

attribute\_part\_2: [attribute4](#attribute4) | [attribute3](#attribute3);

**Description**

These rules will match attribute. As rule attribute has too many sub-rules, these sub-rules are divided into two groups. For the same reason, rule attribute\_part\_1 and attribute\_part\_2 are divided into two groups.

**Syntax**

attribute1: {$cmd::mpMDSearch->CheckAttribute()}? (np | jj | vb)

**Description**

This rule will match attribute with one word. A word will be identified as attribute1 only if the result of function CheckAttribute is true.

**Examples**

year, month

**Syntax**

attribute2: {$cmd::mpMDSearch->CheckAttribute()}? np (vb | np)

**Description**

This rule will match attribute with two words. A phrase will be identified as attribute2 only if the result of function CheckAttribute is true.

**Examples**

call center

**Syntax**

attribute3: {$cmd::mpMDSearch->CheckAttribute()}? np np np

**Description**

This rule will match attribute with three words. A phrase will be identified as attribute3 only if the result of function CheckAttribute is true.

**Syntax**

attribute4: {$cmd::mpMDSearch->CheckAttribute()}? np np np np

**Description**

This rule will match attribute with four words. A phrase will be identified as attribute4 only if the result of function CheckAttribute is true.

**Syntax**

element: [element\_part\_2](#element_part_2) | [element\_part\_1](#element_part_1);

element\_part\_1: [element2](#element2) | [element1](#element1);

element\_part\_2: [element4](#element4) | [element3](#element3);

**Description**

These rules will match element. As rule element has too many sub-rules, these sub-rules are divided into two groups. For the same reason, rule element\_part\_1 and element\_part\_2 are divided into two groups.

**Syntax**

element1: {$cmd::mpMDSearch->CheckElement()}? (cd | nnp | np)

**Description**

This rule will match element with one word. A word will be identified as element1 only if the result of function CheckElement is true.

**Examples**

2013, NY

**Syntax**

element2: {$cmd::mpMDSearch->CheckElement()}? nnp nnp

**Description**

This rule will match element with two words. A phrase will be identified as element2 only if the result of function CheckElement is true.

**Examples**

New York

**Syntax**

element3: {$cmd::mpMDSearch->CheckElement()}? nnp nnp nnp

**Description**

This rule will match element with three words. A phrase will be identified as element3 only if the result of function CheckElement is true.

**Syntax**

element4: {$cmd::mpMDSearch->CheckElement()}? nnp nnp nnp nnp

**Description**

This rule will match element with four words. A phrase will be identified as element4 only if the result of function CheckElement is true.

**Syntax**

graph: [graph\_type2](#graph_type2) | [graph\_type1](#graph_type1);

**Description**

This rule will match graph. Rule graph are divided into two groups.

**Syntax**

graph\_type1: {$cmd::mpMDSearch->CheckGraph()}? (np | vb);

**Description**

This rule will match graph type with one word. A word will be identified as graph\_type1 only if the result of function CheckGraph is true. The graph type word can be configured in GraphType.conf.

**Examples**

trend, trending

**Syntax**

graph\_type2: {$cmd::mpMDSearch->CheckGraph()}? np np;

**Description**

This rule will match graph type with two words. A phrase will be identified as graph\_type2 only if the result of function CheckGraph is true. The graph type phrase can be configured in GraphType.conf.

**Examples**

line graph, time series, pie chart

**Syntax**

exist\_filter: {$cmd::mpMDSearch->CheckFilter()}? [exist\_filter\_name](#exist_filter_name);

exist\_filter\_name: prps np+ | jj np+ | dt np+;

**Description**

These rules will match existing filter. A phrase will be identified as existing filter only if the result of function CheckFilter is true.

**Examples**

my region

**Syntax**

ascend: {$cmd::mpMDSearch->CheckAscend()}? (nn | jj | jjs);

**Description**

This rule will match ascend word. A word will be identified as ascend only if the result of function CheckAscend is true. The ascend word can be configured in KeyWord.conf.

**Examples**

bottom, worst, lowest

**Syntax**

descend: {$cmd::mpMDSearch->CheckDescend()}? (nn | jj | jjs);

**Description**

This rule will match descend word. A word will be identified as descend only if the result of function CheckDescend is true. The descend word can be configured in KeyWord.conf.

**Examples**

top, best, highest

**Syntax**

percent: {$cmd::mpMDSearch->CheckPercent()}? nn;

**Description**

This rule will match percent word. A word will be identified as percent only if the result of function CheckPercent is true. The percent word can be configured in KeyWord.conf.

**Examples**

percent, %

**Syntax**

imperative: {$cmd::mpMDSearch->CheckImperative()}? (vb | nn);

**Description**

This rule will match imperative verb. Some imperative verb’s POS will be detected as noun, such as “show”, so we need to wrap a layer on them. A word will be identified as imperative only if the result of function CheckImperative is true. The percent word can be configured in KeyWord.conf.

**Examples**

give, show

## **Leaf rules**

**Name Description Examples**

wh term started with “wh” what, where

md modal can

vb verb term give, show

op operation term greater, less, <, >, =

cc coordinating conjunction and

np member of noun phrase call, center, year

jj adjective term last

cd number 2012, 1,000,000

inpreposition or subordinating conjunction in, by, with

dt determiner the

prps possessive pronoun my

prp personal pronoun me

nn noun term center, year

jjs adjective, superlative best, worst

nnp proper noun, singular NY, DC

pos possessive ending ‘s

rp particle down

to to to

punct punctuation ?, .

# Antlr Bugs

I’ve found two Antlr bugs when developing NLQ.

1. We need to comment out the code below in antlr3tokenstream.inl.

|  |
| --- |
| //m\_tokens.erase( iter1, iter2 ); |

1. Antlr has bugs on **sub-rule with three alternatives. After we transform the grammar file to cpp and hpp files, we need to change CMDParser.cpp manually.**

**The changes are after the line below:**

|  |
| --- |
| // C:\\Users\\qcheng\\Desktop\\CMD.g:113:60: ( ( exist\_filter )=> exist\_filter | ( attribute ( in )? element )=> inside\_attribute\_filter | ( element )=> attribute\_filter )? |

**We need to make the judgment of exist\_filter, inside\_attribute\_filter, attribute\_filter be parallel.**