

# CoScore Rule Representation Manual

## 1 Basic form

Structured JSON language is used instead of the natural language that describes the scoring rules. It consists of three parts: atoms, combos, and comboMode, as shown in **Example 1**.

### Example 1 Basic form illustration

```
{
  "atoms":{...},
  "combos":{...},
  "comboMode": "ADD"
}
```

Among them, atoms are used to represent the scoring rules in use and are composed of several atomic units; combos, a combined scoring method used to represent scoring rules, consists of several combined units.

## 2 Details

### 2.1 Atoms

It is composed of several atomic units, as in **Example 2**. Each atomic unit holds a digital id as a key, and the corresponding value is a JSON unit, which consists of two parts: the rule type (type) and the rule description (desc).

When an atomic rule is invoked, it returns two values. One item is a **logical value**, representing the hit situation of the rule, and the value range is [True, False]; One item is the **solution value**, representing the numerical value of the rule hit.

Generally, when the logical value is True, the solution value is greater than 0; When the logical value is False, the solution value is 0.

### Example 2 Atoms illustration

```
{
  "atoms":{
    "0":{
      "type": "EM",
      "desc": "Hello"
    }
    "1":{...}
  },
  "combos":{...}
}
```

The normative conventions of type and rule description (desc) are shown in **2.1.1-2.1.4**.

#### 2.1.1 Exact Match (EM)

**2.1.1.1** When the rule type (type) is set to EM, it indicates that the atomic unit is a completely exact matching rule unit.

**2.1.1.2** In the case stipulated in 2.1.1.1, the rule description (desc) must be a string. A string can be a single answer string or multiple answer strings separated by a comma ", ".

**2.1.1.3** Under the circumstances stipulated in 2.1.1.1, return the logical value True if and only if the student's answer string is strictly equal to any one of the answer strings stipulated in 2.1.1.2, and solve for the value 1; Otherwise, return the logical value False and the solution value 0, as in **Example 3**.

**Example 3** EM rule illustration

Rule	Answer	Return value
{ "type": "EM", "desc": "大于,>" }	大于	[True, 1]
	>	[True, 1]
	大于等于	[False, 0]
	不大于	[False, 0]

## 2.1.2 Substring Match (SM)

**2.1.2.1** When the rule type (type) is set to SM, it indicates that the atomic unit is a string-exact matching rule unit.

**2.1.2.2** In the case stipulated in 2.1.2.1, the rule description (desc) must be a string. A string can be a single answer string or multiple answer strings separated by a comma ", ".

**2.1.2.3** Under the circumstances stipulated in 2.1.2.1, for each answer string, a hit rule is recorded once (the value increases by 1) if and only if the student's answer contains its content, as in **Example 4**.

**2.1.2.4** If the answer string contains "|", it means that there are multiple synonyms for this item. If one or more of these items (the word list separated by "|" for each item) are hit in the answer, only one hit rule is recorded (the value is recorded as 1). If the synonyms option is "!" At the beginning, once the word is hit, this answer string will be skipped directly. If the synonyms option starts with "~", it indicates that all the words in the answer string need to be removed before matching the remaining synonyms options.

**2.1.2.5** If the final number of rule hits is 0, the logical value False is returned and the solution value is 0. Otherwise, return the logical value True, and the solution value is the number of rule hits.

**Example 4** SM rule illustration

Rule	Answer	Return value
{ "type": "SM", "desc": "爱,祖国 国家" }	我爱国, 我爱祖国母亲	[True, 2]
	我国	[False, 0]
	我家	[False, 0]
	祖国祖国国家国家	[True, 1]

## 2.1.3 Oneway Proximity (OP)

**2.1.3.1** When the rule type (type) is set to OP, it indicates that the atomic unit is a oneway proximity matching rule unit.

**2.1.3.2** In the case stipulated in 2.1.3.1, the rule description (desc) must be a string. A string must start with a numeric value within the range of (0,1] (referred to as N below) and an English

colon ":". The following part can be a single answer string or multiple answer strings separated by a comma ", ".

**2.1.3.3** Under the circumstances stipulated in 2.1.3.1, if and only if the student's answer contains content with the maximum one-way matching degree  $n \geq N$  of the answer string, return the logical value True and solve for the value n; otherwise, return the logical value False and solve for the value 0, as shown in **Example 5**.

**2.1.3.4** The calculation method of oneway proximity: The dynamic programming method is used to calculate the length of the **maximum common subsequence** of the student's answer and the answer string, and then this length is divided by the length of the answer string to obtain the unidirectional closeness degree.

**Example 5** OP rule illustration

Rule	Answer	Return value
<pre>{   "type": "OP",   "desc": "0.4:绕绕落落回" }</pre>	一二绕三四落五回	[True, 0.60]
	一号二号绕三号四号落	[True, 0.40]
	先回再落	[False, 0]
	顺序是: 绕绕落落回	[True, 1.00]

## 2.1.4 Content Similarity (CS)

**2.1.4.1** When the rule type (type) is set to CS, it indicates that the atomic unit is a similarity rule unit.

**2.1.4.2** The rule description and return situation are the same as 2.1.3.2 and 2.1.3.3. Different from them, the solution of "one-way proximity" is changed to "Jaccard similarity".

## 2.2 Combos

It is composed of several combined units, as in **Example 6**. Each combination unit holds a letter id as a key, and the corresponding value is a JSON unit, which consists of three parts: the combination expression (combo), the benchmark score (score), and the scoring mode (mode).

**Example 6** Combos illustration

```
{
  "rules":{...},
  "combos":{
    "A":{
      "combo": "G(0,T(0)) and G(1,T(0))",
      "score": 5,
      "mode": "logic"
    },
    "B":{...}
  }
}
```

The normative conventions for expressions (combo), benchmark scores (score), and scoring modes (mode) are presented in 2.2.1-2.2.2.

### 2.2.1 Combo

**2.2.1.1 Reference to Students' Answers.** For the  $n$ -th blank of students' answers (starting from 0 as the sequence number), four operators are specified:

**The T operator  $T(n)$**  indicates that the  $n$ -th empty original text (string) is introduced into the expression;

**The L operator  $L(n)$**  represents introducing the length (integer) of the  $n$ -th empty into the expression;

**The Q operator  $Q(n)$**  represents introducing the "whether it is empty" case (Boolean value) of the  $n$ -th empty into the expression;

**The F operator  $F(n)$**  indicates that the  $n$ -th null rotation is converted into a numerical value (floating-point number) and introduced into the expression;

In particular, when  $n$  is "\*",  $T(*)$  indicates that all blanks are merged into one piece of text for processing,  $L(*)$  returns the sum of the lengths of all texts,  $Q(*)$  returns the total number of non-blanks, and  $F(*)$  returns the converted value after merging all blanks into one piece of text.

**2.2.1.2 Invoking Atomic Rules.** Utilizing students' responses to invoke atomic rules, stipulating two types of operators:

**The G operator  $G(K,s)$**  represents the calculation of the answer  $s$  using the atomic rule with id  $K$  and taking the logical value.

**The M operator  $M(K,s)$**  represents calculating the answer  $s$  using the atomic rule with id  $K$  and taking the solution value.

**2.2.1.3 Additional Logical Processing.** Additional logical processing is carried out based on the expression  $f$ , and two operators are specified:

**The U operator  $U(f,C)$**  represents taking the upper bound  $C$  of the value  $f$  and returning it.

**The A operator  $A(a,b,...)$**  represents the number of variables that return true values for any number of inputs.

**The X operator  $X(a,b,...)$**  indicates that for any number of variable inputs, the maximum value is returned.

**2.2.1.4 Legal Symbols.** Besides the above operators, legal symbols also include if else and or not  $() == >= !$  True False  $+ - * /$  number. Among them, the use of "if else" is limited to the single-line expression form of "A if B else C".

**2.2.1.5 Solution of Combined Expressions.** The combined expressions constructed from 2.2.1.1 to 2.2.1.4 have a uniquely determined solution return value, which may be of Boolean type or numerical type.

## 2.2.2 Score and mode

**2.2.2.1** The benchmark score (score) must be a numerical value. It represents the base score obtained when the combined scoring rule takes effect. The specific calculation method of this score needs to be coordinated with the scoring mode (mode).

**2.2.2.2** The scoring mode (mode) must be a string and must be either logic or value.

**2.2.2.3** When the scoring mode (mode) is logic, it indicates a logical scoring rule. At this time,

the combined expression must return a Boolean type. If it returns True, the score corresponding to score will be counted; if it returns False, 0 points will be counted.

**2.2.2.4** When the scoring mode (mode) is value, it indicates the numerical scoring rule. At this time, the combined expression must return the numerical type, and the score is [numerical \*score].

### **2.3 ComboMode**

The comboMode must be "ADD" or "MAX". When the mode is "ADD", the final score of the question is the sum of the scoring results of each scoring rule unit in the combination rule. When the mode is "MAX", the final score of the question is the maximum value of the scoring results of each scoring rule unit in the combination rule. In particular, the total score less than 0 points will be set to 0 points, and the total score greater than 10 points will be set to 10 points.