

## Treebank queries

This document contains all the queries run on the treebanked version of the *Ṛgveda* for the analyses presented in the book *Rigvedic similes: a corpus-based analysis of their forms and functions*.

All queries were written in the Udapi query language (<https://udapi.github.io>).

### Query 0: number of annotated similes

```
cat RV.conllu | udapy util.See node='node.deprel in ("case:stm", "mark:stm") and node.lemma in ("na", "iva", "yathā")' → Result: 2329
```

NB:

- Similes in ṚV 10.106.5-6-7-8 are not included because these verses are left untranslated by most translators;
- Similes in ṚV 10.78.4,5,6ab (for a total of 10 similes) are also not included as these verses are missing from the VTB.

## Chapter 4

### Query 1: N. of predicative similes (no PAR)

```
cat RV.conllu | udapy util.See node='node.deprel in ("mark:stm", "case:stm") and node.lemma in ("na", "iva", "yathā") and node.parent.deprel in ("root", "acl:dpct", "xcomp", "acl", "acl:attr", "amod", "acl:result", "acl:part")'
```

### Query 2: N. of STAND - PAR and PAR - STAND orders in all similes

```
cat RV.conllu | udapy util.See node='node.deprel in ("advcl:cmp", "obl:cmp") and len([x for x in node.children if x.lemma in ("na", "iva", "yathā") and x.deprel in ("mark:stm", "case:stm")]) == 1'
```

### Query 3: N. of STAND - PAR and PAR - STAND orders in simple similes

```
cat RV.conllu | udapy util.See node='node.deprel == "obl:cmp" and len([x for x in node.children if x.lemma in ("na", "iva", "yathā") and x.deprel == "case:stm"]) == 1'
```

### Query 4: N. of STAND - PAR and PAR - STAND orders in similes with gapping

```
cat RV.conllu | udapy util.See node='node.deprel in ("advcl:cmp") and len([x for x in node.children if x.lemma in ("na", "iva", "yathā") and x.deprel == "mark:stm"]) == 1'
```

### Query 5: N. of STAND - PAR and PAR - STAND orders in simple similes with simple STAND

```
cat RV.conllu | udapy util.See node='node.deprel == "obl:cmp" and len([x for x in
node.children if x.lemma in ("na", "iva", "yathā") and x.deprel == "case:stm"]) == 1 and len([x
for x in node.children if x.lemma not in ("na", "iva", "yathā")]) == 0'
```

**Query 6:** N. of STAND - PAR and PAR - STAND orders in simple similes with complex STAND

```
cat RV.conllu | udapy util.See node='node.deprel == "obl:cmp" and len([x for x in
node.children if x.lemma in ("na", "iva", "yathā") and x.deprel == "case:stm"]) == 1 and len([x
for x in node.children if x.lemma not in ("na", "iva", "yathā")]) >= 1'
```

**Query 7:** N. of STAND - PAR and PAR - STAND orders in similes with gapping and simple STAND

```
cat RV.conllu | udapy util.See node='node.deprel == "advcl:cmp" and len([x for x in
node.children if x.lemma in ("na", "iva", "yathā") and x.deprel == "mark:stm"]) == 1 and
len([x for x in node.children if x.lemma not in ("na", "iva", "yathā")]) == 1'
```

**Query 8:** N. of STAND - PAR and PAR - STAND orders in similes with gapping and complex STAND

```
cat RV.conllu | udapy util.See node='node.deprel == "advcl:cmp" and len([x for x in
node.children if x.lemma in ("na", "iva", "yathā") and x.deprel == "mark:stm"]) == 1 and
len([x for x in node.children if x.lemma not in ("na", "iva", "yathā")]) >= 2'
```

## Chapter 6

**Query 9:** Lexemes occurring in the STAND slot (lemma of standard's head)

```
cat RV.conllu | udapy util.Eval node='if node.lemma in ("iva", "na", "yathā") and node.deprel
in ("case:stm", "mark:stm"): count_node.parent.lemma +=1' end='pp(self.count)' Query 10:
lemma in VTB
```

**Query 10:** Absolute frequency of lemmas obtained with Query 9 in the Vedic Treebank:

```
cat RV.conllu | udapy util.Eval node='if node.lemma in ("lemma", "lemma", ...):
count_node.lemma +=1' end='pp(self.count)'
```

NB: The script provided in the repository “Collexeme Analysis < CA\_Simple” subtracts the frequency of a lemma in similes from its absolute frequency in the VTB.

**Query 11:** Total N. of constructions (all nodes that may potentially function as standards of similes)

```
cat RV.conllu | udapy util.See node='node.upos in ("NOUN", "ADJ") and node.parent.deprel
in ("root", "acl", "advcl", "acl:relcl", "acl:dpct", "xcomp", "advcl:temp", "ccomp",
"advcl:dpct", "csubj", "acl:ptcp")'
```

```
cat RV.conllu | udapy util.See node='node.upos == "VERB" and node.feats["VerbForm"] ==
"Part" and node.parent.deprel in ("root", "acl", "advcl", "acl:relcl", "acl:dpct", "xcomp",
"advcl:temp", "ccomp", "advcl:dpct", "csubj", "acl:ptcp")'
```

**Query 12:** Lemmas distinctive to *ná* and *iva* (head of standard)

```
cat RV.conllu | udapy util.Eval node='if node.lemma == "na" and node.deprel in ("case:stm", "mark:stm"): count_node.parent.lemma +=1' end='pp(self.count)'
```

```
cat RV.conllu | udapy util.Eval node='if node.lemma == "iva" and node.deprel in ("case:stm", "mark:stm"): count_node.parent.lemma +=1' end='pp(self.count)'
```

**Query 13:** Forms distinctive to *ná* and *iva* (head of standard)

```
cat RV.conllu | udapy util.Eval node='if node.lemma == "na" and node.deprel in ("case:stm", "mark:stm"): count_node.parent.form +=1' end='pp(self.count)'
```

```
cat RV.conllu | udapy util.Eval node='if node.lemma == "iva" and node.deprel in ("case:stm", "mark:stm"): count_node.parent.form +=1' end='pp(self.count)'
```

**Query 14:** Cases distinctive to *ná* and *iva* (head of standard)

```
cat RV.conllu | udapy util.Eval node='if node.lemma == "na" and node.deprel in ("case:stm", "mark:stm"): count_node.parent.feats["Case"] +=1' end='pp(self.count)'
```

```
cat RV.conllu | udapy util.Eval node='if node.lemma == "iva" and node.deprel in ("case:stm", "mark:stm"): count_node.parent.feats["Case"] +=1' end='pp(self.count)'
```

**Query 15:** Numbers distinctive to *ná* and *iva* (head of standard)

```
cat RV.conllu | udapy util.Eval node='if node.lemma == "na" and node.deprel in ("case:stm", "mark:stm"): count_node.parent.feats["Number"] +=1' end='pp(self.count)'
```

```
cat RV.conllu | udapy util.Eval node='if node.lemma == "iva" and node.deprel in ("case:stm", "mark:stm"): count_node.parent.feats["Number"] +=1' end='pp(self.count)'
```

**Query 16:** Endings and ending patterns distinctive of *ná* and *iva* (< form of the word preceding STM)

```
cat RV.conllu | udapy util.Eval node='if node.lemma == "na" and node.deprel in ("case:stm", "mark:stm"): count_node.prev_node.form +=1' end='pp(self.count)'
```

```
cat RV.conllu | udapy util.Eval node='if node.lemma == "iva" and node.deprel in ("case:stm", "mark:stm"): count_node.prev_node.form +=1' end='pp(self.count)'
```

## Chapter 6

**Query 17:** a. Standard constructed with the parameter

- a. Standard constructed with a finite verb

```
cat RV.conllu | udapy -TM util.Mark node='node.lemma in ("na", "iva", "yathā") and node.deprel in ("case:stm", "mark:stm") and node.parent.upos == "VERB" and node.parent.feats["VerbForm"] == "" | less -R'
```

- b. Standard constructed with any kind of parameter (orphan)

```
cat RV.conllu | udapy -TM util.Mark node='node.lemma in ("na", "iva", "yathā") and
node.deprel in ("case:stm", "mark:stm") and node.parent.deprel == "orphan" | less -R
```

**Query 18:** Standards interrupted by (elements of) the parameter

- a. Standards of simple similes interrupted by the parameter, order x PAR, obl:cmp

```
cat RV.conllu | udapy -TM util.Mark node='node.deprel != "case:stm" and node.parent.deprel
== "obl:cmp" and node.parent.parent.ord > node.ord and node.parent.parent.ord <
node.parent.ord' | less -R
```

- b. standards of simple similes interrupted by the parameter, order obl:cmp, PAR, x

```
cat RV.conllu | udapy -TM util.Mark node='node.deprel != "case:stm" and node.parent.deprel
== "obl:cmp" and node.parent.parent.ord < node.ord and node.parent.parent.ord >
node.parent.ord' | less -R
```

- c. standards of double similes interrupted by the parameter, order advcl:cmp, PAR, x

```
cat RV.conllu | udapy -TM util.Mark node='node.deprel == "orphan" and node.parent.deprel
== "advcl:cmp" and node.parent.parent.ord < node.ord and node.parent.parent.ord >
node.parent.ord' | less -R
```

- d. : standards of double similes interrupted by the parameter, order x, PAR, advcl:cmp

```
cat RV.conllu | udapy -TM util.Mark node='node.deprel == "orphan" and node.parent.deprel
== "advcl:cmp" and node.parent.parent.ord > node.ord and node.parent.parent.ord <
node.parent.ord' | less -R
```

**Query 19:** N. of STAND - PAR and PAR - STAND orders in *yáthā* similes

- a. all *yáthā* similes:

```
cat RV.conllu | udapy util.See node='node.deprel in ("obl:cmp", "advcl:cmp") and
node.feats["VerbForm"] == "" and len([x for x in node.children if x.lemma == "yathā" and
x.deprel in ("mark:stm", "case:stm")]) == 1'
```

- b. with ellipsis:

```
cat RV.conllu | udapy util.See node='node.deprel == "obl:cmp" and node.feats["VerbForm"]
== "" and len([x for x in node.children if x.lemma == "yathā" and x.deprel in ("case:stm")])
== 1'
```

- c. with gapping:

```
cat RV.conllu | udapy util.See node='node.deprel == "advcl:cmp" and
node.feats["VerbForm"] == "" and len([x for x in node.children if x.lemma == "yathā" and
x.deprel in ("mark:stm")]) == 1'
```