Computational tools for the MECORE database

Tomasz Klochowicz

20.06.2024

ILLC, University of Amsterdam

University of Amsterdam

NWO

Institute for Logic, Language and Computation

Table of contents

The bigger picture

D-N Model of Science (Hempel and Oppenheim, 1948)

1. Observation
2. Hypothesis
3. Falsification
4. Explanation

D-N Model of Science (Hempel and Oppenheim, 1948)

1. Observation
    - *know wh- vs. #believe wh-*
2. Hypothesis
3. Falsification
4. Explanation

D-N Model of Science (Hempel and Oppenheim, 1948)

1. Observation
    - *know wh- vs. #believe wh-*
2. Hypothesis
    - All veridical predicates are responsive Egré (2008).
3. Falsification
4. Explanation

D-N Model of Science (Hempel and Oppenheim, 1948)

1. Observation
    - *know wh- vs. #believe wh-*
2. Hypothesis
    - All veridical predicates are responsive Egré (2008).
3. Falsification
    - *Regret, Resent...*
4. Explanation

D-N Model of Science (Hempel and Oppenheim, 1948)

1. Observation
    - *know wh- vs. #believe wh-*
2. Hypothesis
    - All veridical predicates are responsive Egré (2008).
3. Falsification
    - *Regret, Resent...*
4. Explanation
    - 'Whether-complements denote the corresponding true answer' (Egré, 2008, p.20).

MECORE Database

Goals:[1]

- Collect data from 14 languages.
- Develop theoretical hypotheses about [...] clausal embedding.
- Quantitatively evaluate these hypotheses.

[1] https://wuegaki.ppls.ed.ac.uk/mecore/about/

MECORE Database

1. **Observation**: $\leftarrow$ Provides data points for analysis.
2. **Hypothesis**: Develop hypotheses (qualitatively).
3. **Falsification** $\leftarrow$ **quantitative**
4. **Explanation**: Explain the hypotheses.

The focus of this project

1. **Observation** $\leftarrow$ MECORE database as a ‘single’ observation
2. **Hypothesis** $\leftarrow$ quantitative generation
3. **Falsification** $\leftarrow$ **quantitative**
4. **Explanation**: Qualitative analysis.

How to discover (semantic) universals?

The goal

Goal:
Automatically discover universal patterns in the data.

Predicate	Prop A	Prop B	...	class
P_1	1	0	...	1
P_2	1	0	...	1
P_3	1	1	...	0
P_4	0	0	...	0

The goal

Goal:
Automatically discover universal patterns in the data.

Predicate	Prop A	Prop B	...	class
P_1	1	0	...	1
P_2	1	0	...	1
P_3	1	1	...	0
P_4	0	0	...	0

Discovered: All predicates with Property *A* and not property *B* are of class 1

The scope

- The tools can be applied to discover universals of any kind.
- They can be applied to any data set where we expect to find universal patterns.

The scope

- The tools can be applied to discover universals of any kind.
- They can be applied to any data set where we expect to find universal patterns.

For the sake of this talk:
Suppose we are interested in the distinction between **responsive** and **anti-rogative** predicates.

What do we see?

Predicate	veridical	preferential	responsive
know	1	0	1
be unaware	1	0	1
regret	1	1	0
believe	0	0	0

What do we see?

Predicate	veridical	preferential	responsive
know	1	0	1
be unaware	1	0	1
regret	1	1	0
believe	0	0	0

What do we see?

Predicate	veridical	preferential	responsive
know	1	0	1
be unaware	1	0	1
regret	1	1	0
believe	0	0	0

Discovered: All veridical predicates are responsive (Egré, 2008)

What do we see?

Predicate	veridical	preferential	responsive
know	1	0	1
be unaware	1	0	1
regret	1	1	0
believe	0	0	0

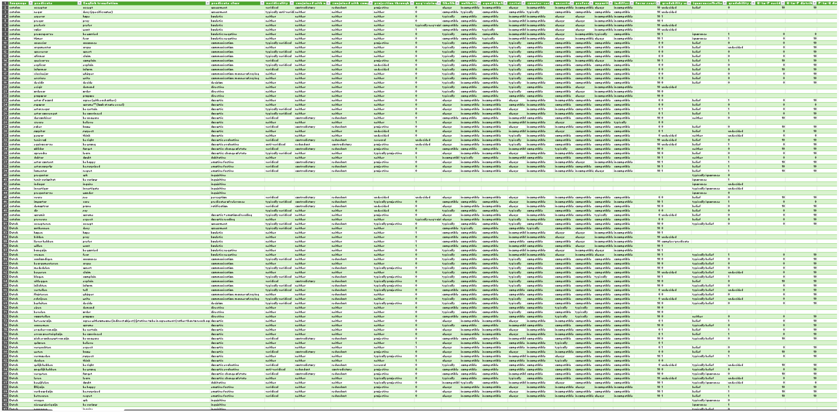
Discovered: All veridical predicates are responsive (Egré, 2008)

What do we see?

Predicate	veridical	preferential	responsive
know	1	0	1
be unaware	1	0	1
regret	1	1	0
believe	0	0	0

Discovered: All veridical [and non-preferential] predicates are responsive (Egré, 2008)

What does the computer see?



How can we make sense of this much data?

Generalisations

The database contains 80 binary semantic properties, which results in more than 6000 potential hypotheses, which consider two properties and almost 500,000, which involve three for each clause-type.

Generalisations

The database contains 80 binary semantic properties, which results in more than 6000 potential hypotheses, which consider two properties and almost 500,000, which involve three for each clause-type.

It is impossible to check them one by one. We can try to analytically pick one, but how can we be sure that this is the most general and accurate hypothesis?

Generalisations

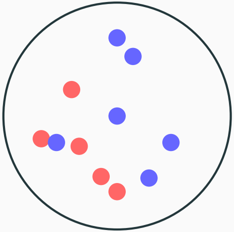
The database contains 80 binary semantic properties, which results in more than 6000 potential hypotheses, which consider two properties and almost 500,000, which involve three for each clause-type.

It is impossible to check them one by one. We can try to analytically pick one, but how can we be sure that this is the most general and accurate hypothesis?

Let’s see a way of checking all the possible generalisations!

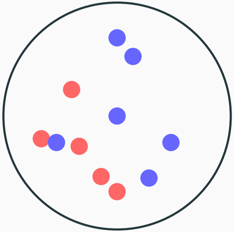
Decision trees

1. Take a set of predicates



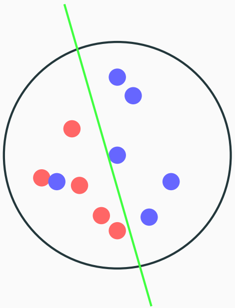
Decision trees

1. Take a set of predicates
2. Pick one property which minimizes impurity.



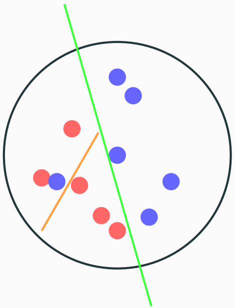
Decision trees

1. Take a set of predicates
2. Pick one property which minimizes impurity.
3. Split the set based on that property



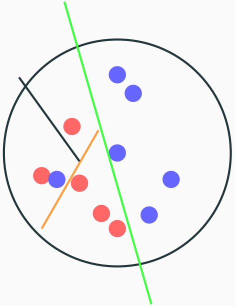
Decision trees

1. Take a set of predicates
2. Pick one property which minimizes impurity.
3. Split the set based on that property
4. Repeat.



Decision trees

1. Take a set of predicates
2. Pick one property which minimizes impurity.
3. Split the set based on that property
4. Repeat.

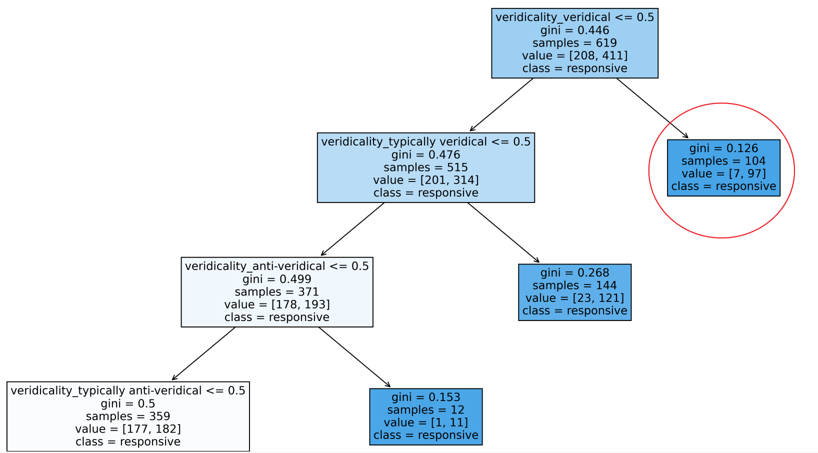


Decision trees

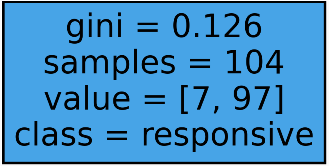
Consider the set of properties: {*Veridicality*} with possible values: *veridical*, *typically veridical*, *typically anti-veridical*, *anti-veridical*.

Decision trees

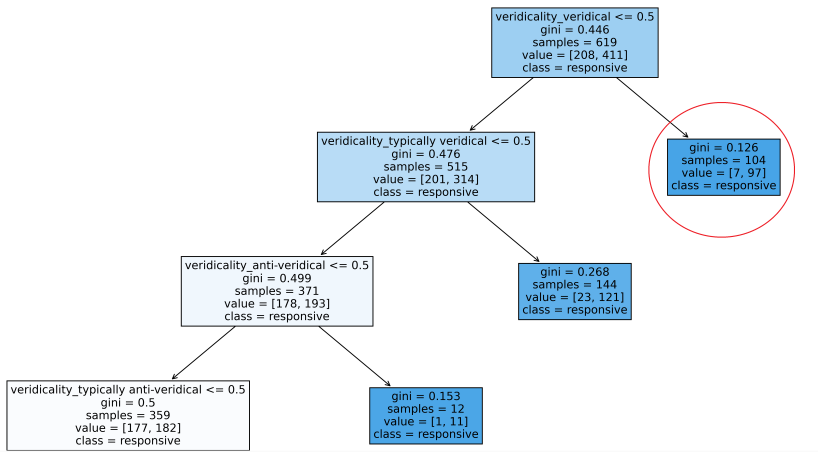
Consider the set of properties: $\{Veridicality\}$ with possible values: *veridical*, *typically veridical*, *typically anti-veridical*, *anti-veridical*.



Decision trees



Decision trees



Exceptions

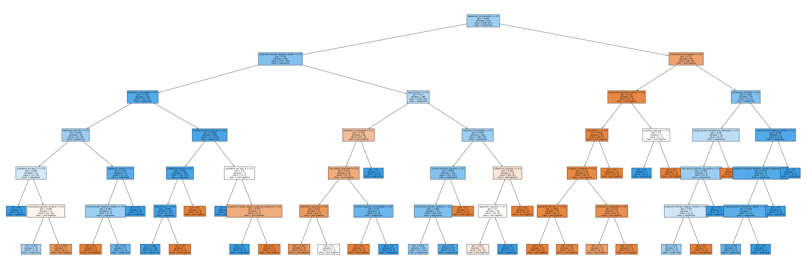
From a decision tree we can extract exceptions:

Language	Predicate	Translation	Veridicality	embedding
German	recht haben	be right	veridical	anti-rogative
Greek	metaniono	regret	veridical	anti-rogative
Hindi	khush	be happy	veridical	anti-rogative
Hindi	hairaan	be surprised	veridical	anti-rogative
Hindi	khed	regret	veridical	anti-rogative
Italian	rimpiangere	regret	veridical	anti-rogative
Polish	żałować	regret	veridical	anti-rogative

Table 1: Exceptions to the veridicality hypothesis

Extracting genralisations

Construct the decision tree from the set of all properties!



Extracting genralisations

Construct the decision tree from the set of all properties!

- Every branch of the tree corresponds to a potential generalisation!
- We check which leaves of the tree are (almost) pure.
- We follow the branch and discover on which properties it splits.
- Issue: The tree only has one root, i.e. all generalisations will contain the root property.
  $\rightsquigarrow$ We will discuss later how we solve it.

Results

Novel Hypothesis: $\mathbb{PNB}$

$\mathbb{PNB}$: <u>All positively preferential predicates which are neutral w.r.t likelihood are anti-rogative.</u> (60 predicates)

Novel Hypothesis: $\mathbb{PNB}$

$\mathbb{PNB}$: <u>All positively preferential predicates which are neutral w.r.t likelihood are anti-rogative.</u> (60 predicates)

Predicate P is **positively preferential** iff sentence $\ulcorner sPc \urcorner$ implies that agent s prefers the complement c over its negation $\neg c$.

Example

The English predicate “hope” is preferential since:

(1) Alfred hopes that Bertrand will leave.
$\rightsquigarrow$ Alfred prefers the possibility that Bertrand will leave over the possibility that Bertrand will stay.

Novel Hypothesis: $\mathbb{PNB}$

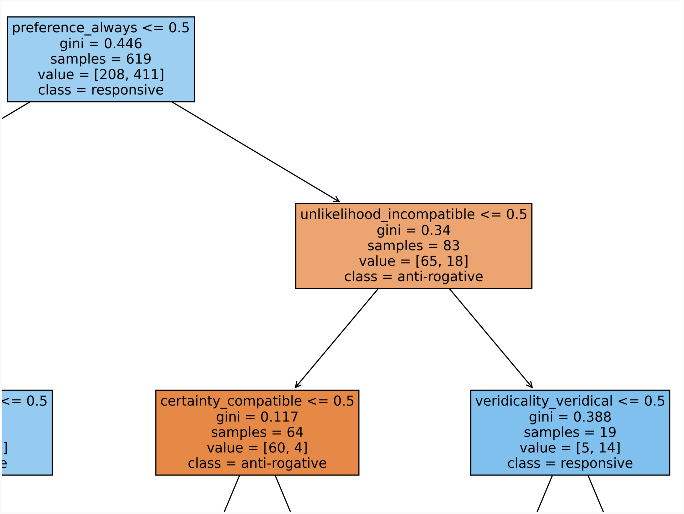
$\mathbb{PNB}$: All positively preferential predicates which are neutral w.r.t likelihood are anti-rogative. (60 predicates)
Predicate P is **neutral w.r.t likelihood** sentence $\ulcorner sPc \urcorner$ neither implies that agent s finds c more likely nor that they find $\neg c$ more likely.

Example
The English predicate “hope” is neutral w.r.t likelihood since:

(2) Alfred hopes that Bertrand will leave.
$\not\rightsquigarrow$ Alfred finds it more/less likely that Bertrand will leave than that Bertrand will stay.

Novel Hypothesis: $\mathbb{PNB}$

$\mathbb{PNB}$: All positively preferential predicates which are neutral w.r.t likelihood are anti-rogative. (60 predicates)

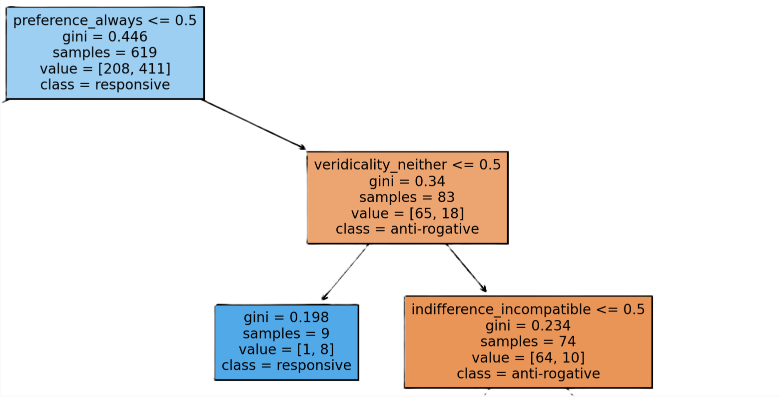


Non-veridical preferential predicates (Uegaki and Sudo, 2019).

All non-veridical and (positive) preferential predicates are anti-rogative.[2]

[2] About positivity see: Klochowicz (2022) or Qing et al. (2024)

All non-veridical and (positive) preferential predicates are anti-rogative.[2]



[2] About positivity see: Klochowicz (2022) or Qing et al. (2024)

Veridical Preferential predicates (Romero, 2015; Uegaki and Sudo, 2019)

$$\|[\text{be happy}_C]\|^w$$
$$= \lambda Q \lambda x : \exists p \in Q[p(w) \wedge B_w(x, p)].$$
$$\exists p' \in Q \left[\begin{array}{c} p'(w) \wedge B_w(x, p') \wedge \\ Pref_w(x, p') > \theta(\{Pref_w(x, p'') \mid p'' \in C\}) \end{array} \right]$$

Veridical Preferential predicates (Romero, 2015; Uegaki and Sudo, 2019)

$|[\text{be happy}_C]|^w$

$$= \lambda Q \lambda x : \exists p \in Q[p(w) \wedge B_w(x, p)].$$

$$\exists p' \in Q \left[\begin{array}{c} p'(w) \wedge B_w(x, p') \wedge \\ Pref_w(x, p') > \theta(\{Pref_w(x, p'') \mid p'' \in C\}) \end{array} \right]$$

- is defined only if:
  - there is $p \in Q$ s.t. p is true at w.
  - x believes p in w.

Veridical Preferential predicates (Romero, 2015; Uegaki and Sudo, 2019)

$$
\begin{aligned}
&|[\text{be happy}_C]|^w \\
&\quad = \lambda Q \lambda x : \exists p \in Q[p(w) \wedge B_w(x, p)]. \\
&\qquad\qquad\qquad \exists p' \in Q \left[\begin{array}{c} p'(w) \wedge B_w(x, p') \wedge \\ Pref_w(x, p') > \theta(\{Pref_w(x, p'') \mid p'' \in C\}) \end{array} \right]
\end{aligned}
$$

- is defined only if:
  - there is $p \in Q$ s.t. p is true at w.
  - x believes p in w.
- is true if there is $p' \in Q : p'$ is true, x believes in p' and x has a preference for p' higher than threshold θ.

Non-Veridical Preferential predicates (Uegaki and Sudo, 2019)

$$|[\text{hope}_C]|^w$$
$$= \lambda Q \lambda x : \cancel{\exists p \in Q[p(w) \wedge B_w(x,p)]}.$$
$$\exists p' \in Q \begin{bmatrix} \cancel{p'(w) \wedge B_w(x,p') \wedge} \\ Pref_w(x,p') > \theta(\{Pref_w(x,p'') \mid p'' \in C\}) \end{bmatrix}$$

- is defined only if:
  - ~~there is $p \in Q$ s.t. p is true at w.~~
  - ~~x believes p in w.~~
- is true if there is $p' \in Q$: ~~p' is true, x believes in p' and~~ x has a preference for p' higher than threshold θ.

Likelihood implying preferential predicates

$|[\text{îrîgîîra}_C]|^w$

$$= \lambda Q \lambda x : \exists p \in Q[\cancel{p(w)} \wedge B_w(x, p)].$$

$$\exists p' \in Q \left[\begin{array}{c} \cancel{p'(w) \wedge} B_w(x, p') \wedge \\ Pref_w(x, p') > \theta(\{Pref_w(x, p'') \mid p'' \in C\}) \end{array} \right]$$

- is defined only if:
  - there is $p \in Q$ s.t. ~~p is true at w~~
  - x believes p in w.

Likelihood implying preferential predicates

$|[\text{îrîgĩĩra}_C]|^w$
$= \lambda Q \lambda x : \exists p \in Q[\cancel{p(w)} \wedge B_w(x, p)].$

$$\exists p' \in Q \left[\begin{array}{c} \cancel{p'(w)} \wedge B_w(x, p') \wedge \\ Pref_w(x, p') > \theta(\{Pref_w(x, p'') \mid p'' \in C\}) \end{array} \right]$$

- is defined only if:
  - there is $p \in Q$ s.t. ~~*p* is true at *w*~~
  - x believes p in w.
- is true if there is $p' \in Q$: ~~p' is true~~, x believes in p' and x has a preference for p' higher than threshold θ.

(*îrîgĩĩra* translates from Kĩĩtharaka as *to hope*)

Likelihood neutral veridical preferential predicates

- is defined only if:
  - there is $p \in Q$ s.t. p is true at w.
  - ~~x believes p in w~~.
- is true if there is $p' \in Q$: p' is true, ~~x believes in p'~~ and x has a preference for p' higher than threshold θ.

There are no such predicates in the database.

Likelihood neutral veridical preferential predicates

$|[\text{will be happy}_C]|^w$

$= \lambda Q \lambda x : \exists p \in Q[p(w) \cancel{\wedge B_w(x,p)}].$

$$\exists p' \in Q \left[\begin{array}{c} p'(w) \cancel{\wedge B_w(x,p')} \wedge \\ Pref_w(x,p') > \theta(\{Pref_w(x,p'') \mid p'' \in C\}) \end{array} \right]$$

- is defined only if:
  - there is $p \in Q$ s.t. p is true at w.
  - ~~x believes p in w~~.
- is true if there is $p' \in Q : p'$ is true, ~~x believes in p'~~ and x has a preference for p' higher than threshold θ.

There are no such predicates in the database.

Likelihood neutral veridical preferential predicates

(3) John will be happy that Mary won the race.

(4) John will be happy (about) who won the race.

- is defined only if:
  - there is $p \in Q$ s.t. p is true at w.
  - ~~x believes p in w~~.
- is true if there is $p' \in Q : p'$ is true, ~~x believes in p'~~ and x has a preference for p' higher than threshold θ.

There are no such predicates in the database.

Exceptions

However, the database still contains 4 counterexamples to both Uegaki and Sudo's proposal and $\mathbb{PNB}$: Greek *προτείνω* (*suggest*), Hebrew *lehaadíf* (*prefer*), Kĩĩtharaka *menyeera* (*care*) and Turkish *um* (*hope*), which remain unaccounted for.

[3] This predicate seems to be veridical e.g.: #*Martin est heureux que Candice fasse le cours de syntaxe le mardi mais elle ne le fera pas.*

Exceptions

However, the database still contains 4 counterexamples to both Uegaki and Sudo's proposal and $\mathbb{PNB}$: Greek *προτείνω* (*suggest*), Hebrew *lehaadíf* (*prefer*), Kĩĩtharaka *menyeera* (*care*) and Turkish *um* (*hope*), which remain unaccounted for.

Moreover, we 'loose' explanation for four predicates, which are non-veridical, imply likelihood, but are anti-rogative: French *être heureux* (be happy)[3], Hindi *apeksha* (expect), Japanese*konom-u* (prefer) and Mandarin Chinese *gaoxing* (be happy),

[3]This predicate seems to be veridical e.g.: *#Martin est heureux que Candice fasse le cours de syntaxe le mardi mais elle ne le fera pas.*

Novel hypothesis: $\mathbb{UNG}$

$\mathbb{UNG}$: All predicates which always imply uncertainty and are not gradable are anti-rogative.

Novel hypothesis: $\mathbb{UNG}$

$\mathbb{UNG}$: All predicates which always imply uncertainty and are not gradable are anti-rogative.

Predicate P is **uncertainty implying** iff sentence $\ulcorner sPc \urcorner$ neither implies that agent s is uncertain whether c is the case or not.

Example
The English predicate “suspect” is uncertainty implying since:

(5) Alfred suspects that Bertrand will leave.
$\rightsquigarrow$ Alfred is uncertain whether Bertrand will leave.

Novel hypothesis: $\mathbb{UNG}$

$\mathbb{UNG}$: <u>All predicates which always imply uncertainty and are not gradable are anti-rogative.</u>

Predicate *P* is **non-gradable** if it cannot be modified by expressions like ‘strongly’ or ‘very much’

Example
The English predicate “know” is non gradable:

(6) #Alfred knows very much that Bertrand will leave.

(7) #Alfred strongly knows that Bertrand will leave.

Novel hypothesis: $\mathbb{UNG}$

$\mathbb{UNG}$: All predicates which always imply uncertainty and are not gradable are anti-rogative.

There are two counterexamples to this hypothesis. Polish verb *podejrzewać* (*suspect*) and Italian *dimenticarsi* (*forget*). A more detailed analysis reveals that the first verb has gradable uses (Pęzik, 2012), and the second is compatible with counter-certainty.

Interim conclusions

- Exceptions:
  - The margin for error.
  - The most interesting data points to study.
- Design-dependency:
  - The design of the MECORE database is theory-driven: we can expect results similar to the theory.
  - Bigger chance of novel findings in unexplored domains (e.g. distribution of NPIs).
- Places for theoretical research: design, explanation.

Technical details

Baseline

1. Take all conjunctions of properties size < n.
2. For each conjunction check if any combination of values implies a label.
3. Report all the discovered hypotheses.

Issues:

- Running time: For n = 3 it is over 20 minutes.
- Many redundant hypotheses (almost 9000 in total)

From trees to forests

1. Take all sets of properties size n.
2. For each set construct a decision tree.
3. Report all the discovered hypotheses.

- For $n = 3$ it is around 40 seconds (not 20 min).
- Returns 1360 hypotheses (not 8000) some are still redundant

Cutting trees

1. Construct a tree from the set of all properties
2. Report all the discovered hypotheses.
3. Eliminate the root property.
4. Repeat

- Feature: Very fast (a couple of seconds)
- Issue: Not exhaustive.

Quality of hypotheses

Average performance on randomly generated data:

- Number of predicates in a hypothesis: 23
- Number of exceptions: 11
- Ratio exceptions/predicates: 50%

Quality of hypotheses

Average performance on randomly generated data:

- Number of predicates in a hypothesis: 23
- Number of exceptions: 11
- Ratio exceptions/predicates: 50%

Average performance on actual data:

- Number of predicates in a hypothesis: 26
- Number of exceptions: 5
- Ratio exceptions/predicates: 26%

https://github.com/TJKlochowicz/Mecore_analysis_tools

Email me: *t.j.klochowicz@uva.nl*

https://github.com/TJKlochowicz/Mecore_analysis_tools

Email me: *t.j.klochowicz@uva.nl*

Thank you!

References

Egré, P. (2008). Question-embedding and factivity. Grazer Philosophische Studien, 77(1):85–125.

Hempel, C. G. and Oppenheim, P. (1948). Studies in the logic of explanation. Philosophy of science, 15(2):135–175.

Klochowicz, T. (2022). Investigating the properties of clause-embedding predicates in polish. In Degano, M., Roberts, T., Sbardolini, G., and Schouwstra, M., editors, Proceedings of the 23rd Amsterdam Colloquium, pages 166–172.

Pęzik, P. (2012). Wyszukiwarka PELCRA dla danych NKJP. Narodowy Korpus Języka Polskiego. Przepiórkowski A., Bańko M., Górski R., Lewandowska-Tomaszczyk B. (ed.).

Qing, C., Ozyıldız, D., Roelofsen, F., Romero, M., and Uegaki, W. (2024). When can non-veridical preferential attitude predicates take questions? Lingbuzz: https://ling.auf.net/lingbuzz/008113.

Romero, M. (2015). Surprise-predicates, strong exhaustivity and alternative questions. In Semantics and linguistic theory, volume 25, pages 225–245.

Uegaki, W. and Sudo, Y. (2019). The *hope-wh puzzle. Natural Language Semantics, 27(4):323–356.