What embedded sentences do

Finding new generalizations in the sea of big data

Tom Roberts

Utrecht University

EGG, 8 August 2025

What was this class about?

We examined many clausal-embedding predicates, with a particular eye towards **why predicates embed the sorts of clauses they do**

What was this class about?

We examined many clausal-embedding predicates, with a particular eye towards **why predicates embed the sorts of clauses they do**

Some (imperfect) generalizations:

- If neg-raising, then anti-rogative
- If veridical, then responsive

How should we generate/test *new* hypotheses about clausal embedding (or anything else)?

How should we generate/test *new* hypotheses about clausal embedding (or anything else)?

- Typical in linguistics: germ of hypothesis formed by observing language use in the wild
- Less typical (in semantics): computationally generating numerous hypotheses and seeing which best fit the data

Judgments in the clausal embedding domain are complex, so data is often scarce

Today: Two approaches to scaling up research on clausal-embedding: large-scale acceptability studies and cross-linguistic databases

MegaAttitude

Website: `https://megaattitude.io/`

Family of large-scale acceptability studies of English attitude predicates

MegaAttitude

Website: `https://megaattitude.io/`

Family of large-scale acceptability studies of English attitude predicates

- Large-scale: 1000+ sentences, 50+ syntactic frames, 5-ish observations per predicate per frame
- Also data about neg-raising, temporal orientation, veridicality, etc.

MegaAttitude

Website: `https://megaattitude.io/`

Family of large-scale acceptability studies of English attitude predicates

- Large-scale: 1000+ sentences, 50+ syntactic frames, 5-ish observations per predicate per frame
- Also data about neg-raising, temporal orientation, veridicality, etc.

Why use large datasets

Useful for lexicon-scale reasoning, e.g. identifying lexical gaps

Why use large datasets

Useful for lexicon-scale reasoning, e.g. identifying lexical gaps

Example: stative *contrafactive* predicates, which presuppose their complement is false, are argued to be rare/nonexistent

(Holton 2017; Strohmaier & Wimmer 2022, 2023; Strohmaier 2025; Glass 2025; Sander 2025; Roberts & Özyıldız 2025)

Why use large datasets

Useful for lexicon-scale reasoning, e.g. identifying lexical gaps

Example: stative *contrafactive* predicates, which presuppose their complement is false, are argued to be rare/nonexistent
(Holton 2017; Strohmaier & Wimmer 2022, 2023; Strohmaier 2025; Glass 2025; Sander 2025; Roberts & Özyıldız 2025)

(1) a. Márta knows that it's raining.
Presupposed: It's raining.
b. Márta *shknows* that it's raining.
Presupposed: It's not raining. (Unattested)

Why use large datasets

Useful for lexicon-scale reasoning, e.g. identifying lexical gaps

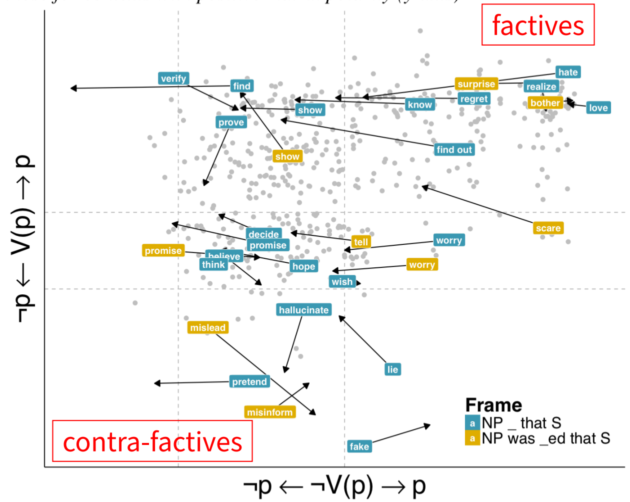
Example: stative *contrafactive* predicates, which presuppose their complement is false, are argued to be rare/nonexistent
(Holton 2017; Strohmaier & Wimmer 2022, 2023; Strohmaier 2025; Glass 2025; Sander 2025; Roberts & Özyıldız 2025)

(1) a. Márta knows that it's raining.
Presupposed: It's raining.
b. Márta *shknows* that it's raining.
Presupposed: It's not raining. (Unattested)

Proving non-existence is hard; broad sampling of evidence which fails to support existence is often the best we can do

No contra-factives in the English lexicon

(14) *Normalized responses for contexts with negative matrix polarity (x-axis) against those for contexts with positive matrix polarity (y-axis)*



Limitations of MegaAttitude

Scaling acceptability judgments comes with costs:

- At the level of individual verbs/frames, data is very noisy
  - More useful for lexicon-scale generalizations
- Frames are semantically low-content (*Someone V that something was true*)

Limitations of MegaAttitude

Scaling acceptability judgments comes with costs:

- At the level of individual verbs/frames, data is very noisy
    - More useful for lexicon-scale generalizations
- Frames are semantically low-content (*Someone V that something was true*)
- Not easily applicable to low-resource language contexts

Limitations of MegaAttitude

Scaling acceptability judgments comes with costs:

- At the level of individual verbs/frames, data is very noisy
  - More useful for lexicon-scale generalizations
- Frames are semantically low-content (*Someone V that something was true*)
- Not easily applicable to low-resource language contexts
- Not cheap (you need to pay a lot of people to do these judgments)

Limitations of MegaAttitude

Scaling acceptability judgments comes with costs:

- At the level of individual verbs/frames, data is very noisy
  - More useful for lexicon-scale generalizations
- Frames are semantically low-content (*Someone V that something was true*)
- Not easily applicable to low-resource language contexts
- Not cheap (you need to pay a lot of people to do these judgments)

MECORE database

Goal: Cross-linguistic database of expert judgments of properties of CE predicates

MECORE database

Goal: Cross-linguistic database of expert judgments of properties of CE predicates

Data: Table of judgments (does verb x have property y) and extensive qualitative surveys for 50-ish verbs

MECORE database

Goal: Cross-linguistic database of expert judgments of properties of CE predicates

Data: Table of judgments (does verb x have property y) and extensive qualitative surveys for 50-ish verbs

Languages: Catalan, Dutch, English, French, German, Greek, Hebrew, Hindi, Italian, Japanese, Kĩĩtharaka, Mandarin, Polish, Spanish, Swedish and Turkish.

Core Team: Ciyang Qing, Floris Roelofsen, Maribel Romero, Wataru Uegaki, Deniz

MECORE database

Goal: Cross-linguistic database of expert judgments of properties of CE predicates

Data: Table of judgments (does verb x have property y) and extensive qualitative surveys for 50-ish verbs

Languages: Catalan, Dutch, English, French, German, Greek, Hebrew, Hindi, Italian, Japanese, Kĩĩtharaka, Mandarin, Polish, Spanish, Swedish and Turkish.

Core Team: Ciyang Qing, Floris Roelofsen, Maribel Romero, Wataru Uegaki, Deniz

Website: `https://wuegaki.ppls.ed.ac.uk/mecore/mecore-databases/`

Extracting patterns from MECORE

Overall goal: Predict a target value (e.g., being responsive) given a conjunction of properties

Extracting patterns from MECORE

Overall goal: Predict a target value (e.g., being responsive) given a conjunction of properties

- We want to efficiently find the best way to predict outcomes

We can use familiar machine learning algorithms to do this.

Extracting patterns from MECORE

Overall goal: Predict a target value (e.g., being responsive) given a conjunction of properties

- We want to efficiently find the best way to predict outcomes

We can use familiar machine learning algorithms to do this.

Tomasz Klochowicz has made some tools to analyze MECORE specifically, available at

`https://github.com/TJKlochowicz/Mecore_analysis_tools`

Structure of MECORE

Data in MECORE consists of matrices of verbs and properties

- Properties: neg-raising, likelihood that embedded *p* is true, etc.
- Values of property variables are categorical

Structure of MECORE

Data in MECORE consists of matrices of verbs and properties

- Properties: neg-raising, likelihood that embedded *p* is true, etc.
- Values of property variables are categorical
  - neg-raising: 0 or 1
  - veridicality: always, typically, typically anti-veridical, anti-veridical

Structure of MECORE

Data in MECORE consists of matrices of verbs and properties

- Properties: neg-raising, likelihood that embedded *p* is true, etc.
- Values of property variables are categorical
  - neg-raising: 0 or 1
  - veridicality: always, typically, typically anti-veridical, anti-veridical
- Not all properties are relevant for every language (e.g. mood)

Computers are good, actually

We are doing a classification problem: ‘find the best label for x given data y’

Goal: Identify properties of attitude reports that predict verbal properties (e.g. being responsive)

Extracting patterns from a big dataset is difficult by hand.

Computers are good, actually

We are doing a classification problem: ‘find the best label for x given data y’

Goal: Identify properties of attitude reports that predict verbal properties (e.g. being responsive)

Extracting patterns from a big dataset is difficult by hand.

- ‘Hypothesis’: values of (combinations of) variables that predictably produce a particular outcome

Computers are good, actually

We are doing a classification problem: 'find the best label for x given data y'

Goal: Identify properties of attitude reports that predict verbal properties (e.g. being responsive)

Extracting patterns from a big dataset is difficult by hand.

- 'Hypothesis': values of (combinations of) variables that predictably produce a particular outcome
- If we have *n* binary variables as potential predictors, there are $n^2 - n$ combinations of two variables to test
  - 80 variables: 6320 possible hypotheses
  - All combinations of 3 variables: nearly 500k
  - How do we find the 'good' hypotheses?

Computers are good, actually

We are doing a classification problem: ‘find the best label for x given data y’

Goal: Identify properties of attitude reports that predict verbal properties (e.g. being responsive)

Extracting patterns from a big dataset is difficult by hand.

- ‘Hypothesis’: values of (combinations of) variables that predictably produce a particular outcome
- If we have *n* binary variables as potential predictors, there are $n^2 - n$ combinations of two variables to test
  - 80 variables: 6320 possible hypotheses
  - All combinations of 3 variables: nearly 500k
  - How do we find the ‘good’ hypotheses?

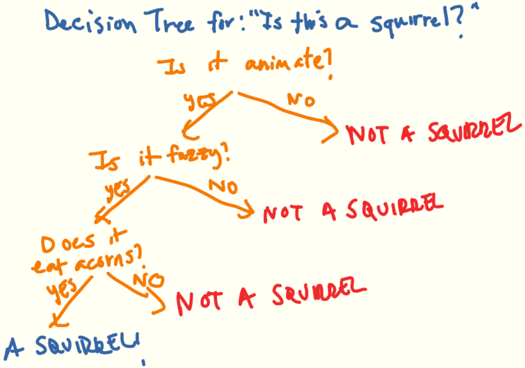
Decision trees

Decision tree learning: A relatively efficient way to find reasonable hypotheses

Decision trees

Decision tree learning: A relatively efficient way to find reasonable hypotheses

Decision trees are essentially flowcharts:



Finding the best decision trees

How do we uncover the decision trees that give us the most useful generalizations?

Finding the best decision trees

How do we uncover the decision trees that give us the most useful generalizations?

Suppose we have a set of datapoints (in *n*-dimensional space), each of which corresponds to a verb

- Each datapoint has a label (say, responsive or non-responsive)

Finding the best decision trees

How do we uncover the decision trees that give us the most useful generalizations?

Suppose we have a set of datapoints (in *n*-dimensional space), each of which corresponds to a verb

- Each datapoint has a label (say, responsive or non-responsive)
- Position in space is determined by values of predictor variables

Finding the best decision trees

How do we uncover the decision trees that give us the most useful generalizations?

Suppose we have a set of datapoints (in *n*-dimensional space), each of which corresponds to a verb

- Each datapoint has a label (say, responsive or non-responsive)
- Position in space is determined by values of predictor variables
- Find a property that, when you divide data by values wrt that property, groups together the most observations with the same label
    - Essentially: draw a straight line through the graph

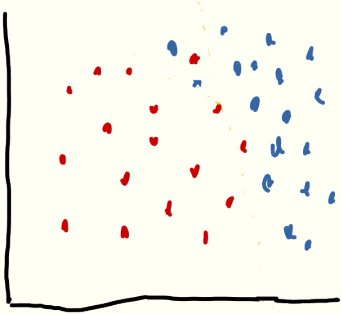
Finding the best decision trees

How do we uncover the decision trees that give us the most useful generalizations?

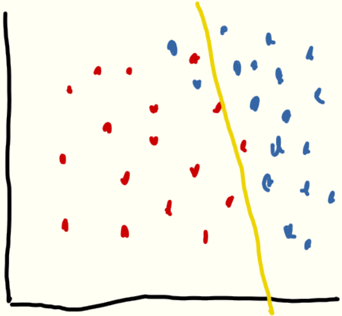
Suppose we have a set of datapoints (in *n*-dimensional space), each of which corresponds to a verb

- Each datapoint has a label (say, responsive or non-responsive)
- Position in space is determined by values of predictor variables
- Find a property that, when you divide data by values wrt that property, groups together the most observations with the same label
  - Essentially: draw a straight line through the graph
- There might be many hypotheses which result in a halfway useful tree (we may not want to consider globally optimal choices only)

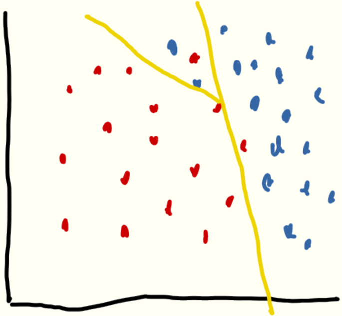
Decision tree: visualized



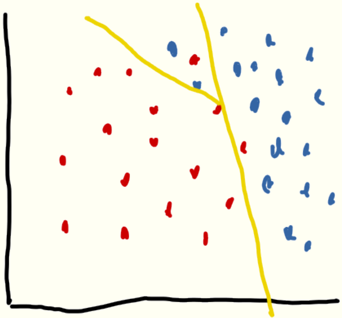
Decision tree: visualized



Decision tree: visualized



Decision tree: visualized



Problems to be addressed: overfitting, pruning redundant branches,...

Benefits of a classifier

Extract a minimal set of properties that best explains some pattern in the data

Benefits of a classifier

Extract a minimal set of properties that best explains some pattern in the data

- $\Rightarrow$ Allows for development of new hypotheses and validation of old ones

Identify counterexamples to generalizations: good for deciding where to follow-up

Benefits of a classifier

Extract a minimal set of properties that best explains some pattern in the data

$\Rightarrow$ Allows for development of new hypotheses and validation of old ones

Identify counterexamples to generalizations: good for deciding where to follow-up

Some hypotheses developed by this approach (Klochowicz 2024):

(2) All positively preferential predicates which are neutral w.r.t likelihood (e.g. *hope*) are anti-rogative.

Benefits of a classifier

Extract a minimal set of properties that best explains some pattern in the data

$\Rightarrow$ Allows for development of new hypotheses and validation of old ones

Identify counterexamples to generalizations: good for deciding where to follow-up

Some hypotheses developed by this approach (Klochowicz 2024):

(2) All positively preferential predicates which are neutral w.r.t likelihood (e.g. *hope*) are anti-rogative.

(3) All predicates which always imply uncertainty and are not gradable (e.g. *suspect*) are anti-rogative.

Limitations

Database is theory-driven

- We have some theoretically-informed ideas about what properties we include to begin with
- Perhaps we might find novel patterns if we broadened the empirical scope: more verbs, different properties

Limitations

Database is theory-driven

- We have some theoretically-informed ideas about what properties we include to begin with
- Perhaps we might find novel patterns if we broadened the empirical scope: more verbs, different properties

Database is labor-intensive

- Populating the database for a given language requires both a lot of time and linguistic expertise
- Upside: we have access to a greater variety of languages than a MegaAttitude-like big data approach

The end

Clausal-embedding is a rich and complex topic.

- Predicates vary in their ability to embed clauses of different types, and clauses in embedded contexts behave quite differently from matrix ones
- There are many interesting correlations between properties of CE predicates and properties of their complements
- We're limited by the pool of languages that have been investigated so far

⋆ ⋆ ⋆If you're gotten anything out of the last two weeks of courses on sentences/clause embedding and you are at any point in the future interested in:

- Talking about a project you're working on
- Getting feedback on an idea or spitballing
- Collaborating on something of mutual interest
- Anything else that seems like it belongs in this list

Send me an email! t.d.h.roberts@uu.nl.