Computational Complexity and Linguistic Theory

Guy Emerson

What I'll Cover

- Turing Completeness of Unification
- Complexity in Phonology
- Judicious Incoherence

What I'll Cover

- Turing Completeness of Unification
  - Powerful formalism for developing theory
- Complexity in Phonology
- Judicious Incoherence

What I'll Cover

- Turing Completeness of Unification
  - Powerful formalism for developing theory
- Complexity in Phonology
  - Constrained theory
- Judicious Incoherence

What I’ll Cover

- Turing Completeness of Unification
  - Powerful formalism for developing theory
- Complexity in Phonology
  - Constrained theory
- Judicious Incoherence
  - Constraints on human behaviour

Turing Completeness of Unification

- Bender and Emerson (2021): In HPSG, “the formalism is stable, even as the theory develops”

Turing Completeness of Unification

- Bender and Emerson (2021): In HPSG, “the formalism is stable, even as the theory develops”
- Computation types and wrapper types (including append-lists) do not change the formalism

Timeline

- 2017 Diff-list appends, list copying

Timeline

- 2017 Diff-list appends, list copying
- 2018 Berthold: this looks fun

Timeline

- 2017 Diff-list appends, list copying
- 2018 Berthold: this looks fun
- 2019 List appends, Turing completeness

Timeline

- 2017 Diff-list appends, list copying
- 2018 Berthold: this looks fun
- 2019 List appends, Turing completeness
- 2020 Olga: this looks fun

Timeline

- 2017 Diff-list appends, list copying
- 2018 Berthold: this looks fun
- 2019 List appends, Turing completeness
- 2020 Olga: this looks fun
- 2021 More careful implementation

Timeline

- 2017 Diff-list appends, list copying
- 2018 Berthold: this looks fun
- 2019 List appends, Turing completeness
- 2020 Olga: this looks fun
- 2021 More careful implementation
- 2023 Dan: this looks fun

Turing Completeness of Unification

- Separating formalism from theory: new applications can be surprising
- Comments welcome on draft paper!
  `https://www.cl.cam.ac.uk/~gete2/wrapper.pdf`

Complexity in Phonology

- Phonological processes widely believed to be finite-state (can be computed in linear time)

Complexity in Phonology

- Phonological processes widely believed to be finite-state (can be computed in linear time)
- Lamont (2023): Optimality Theory is Turing-complete

Complexity in Phonology

- Phonological processes widely believed to be finite-state (can be computed in linear time)
- Lamont (2023): Optimality Theory is Turing-complete
- Hao (2024): Single-tape Optimality Theory is PSPACE-complete (at least as hard as NP)

Complexity in Phonology

- Phonological processes widely believed to be finite-state (can be computed in linear time)
- Lamont (2023): Optimality Theory is Turing-complete
- Hao (2024): Single-tape Optimality Theory is PSPACE-complete (at least as hard as NP)
- Emerson and Lamont (in prep.): Single-tape Harmonic Grammar is finite-state

Optimality Theory

- Underlying phonological representation is distinct from surface realisation
  - e.g. English plural *-s*: [s], [z], [əz]

Optimality Theory

- Underlying phonological representation is distinct from surface realisation
  - e.g. English plural *-s*: [s], [z], [əz]
- A grammar is a set of constraints (which may conflict), and a ranking of those constraints

Optimality Theory

- Underlying phonological representation is distinct from surface realisation
  - e.g. English plural *-s*: [s], [z], [əz]
- A grammar is a set of constraints (which may conflict), and a ranking of those constraints
- Computational problem: given a grammar and an input, what is the optimal output?

Harmonic Grammar

- Underlying phonological representation is distinct from surface realisation
  - e.g. English plural *-s*: [s], [z], [əz]
- A grammar is a set of constraints (which may conflict), and a **weighting** of those constraints
- Computational problem: given a grammar and an input, what is the optimal output?

Harmonic Grammar

- Can use a Viterbi algorithm
  - At each step, sufficient to consider a finite set of candidates

Harmonic Grammar

- Can use a Viterbi algorithm
  - At each step, sufficient to consider a finite set of candidates
- Relies on comparing candidates as a single score

Harmonic Grammar

- Can use a Viterbi algorithm
  - At each step, sufficient to consider a finite set of candidates
- Relies on comparing candidates as a single score
  - In contrast, Optimality Theory requires comparison for each constraint

Harmonic Grammar vs. Optimality Theory

- Theoretical commitments very similar
  - Explain complex phenomena as interactions between simple constraints
- Computational behaviour very different
  - Finite-state vs. PSPACE-complete

10 Years of My Semantics Research…

- Truth is useful

- Truth is painful

10 Years of My Semantics Research…

- Truth is useful
  - A truth-conditional model can learn and generalise meanings in a more human-like way
- Truth is painful

10 Years of My Semantics Research...

- Truth is useful
  - A truth-conditional model can learn and generalise meanings in a more human-like way
- Truth is painful
  - A truth-conditional model is intractable with respect to dimensionality of feature space

Judicious Incoherence

- A computational system cannot be all three:
  - Expressive
  - Tractable
  - Coherent

Judicious Incoherence

- A computational system cannot be all three:
  - Expressive
  - Tractable
  - Coherent
- Human cognition is expressive and tractable, so cannot be coherent

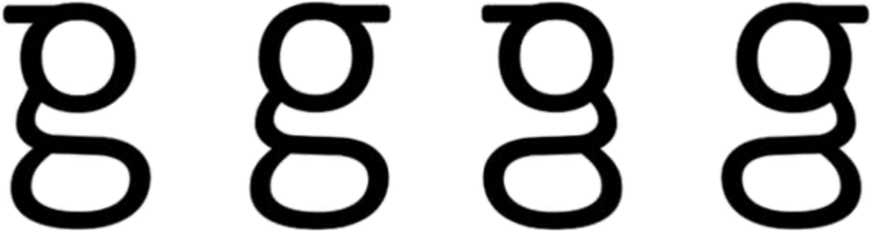
Bayesian Coherence

- Given $p(x, y)$, this defines $p(x|y)$ and $p(y|x)$

Bayesian Coherence

- Given $p(x, y)$, this defines $p(x|y)$ and $p(y|x)$
- Bayesian inference is #P-complete (at least as hard as NP-complete), even in restricted settings and even when approximated (Roth, 1996)

Example: Looptail g



Example: Looptail g

- Recognised easily

- Produced with difficulty or not at all (Wong et al., 2018)

Example: Looptail g

- Recognised easily
  - $p(class|shape)$ accurate
- Produced with difficulty or not at all (Wong et al., 2018)

Example: Looptail g

- Recognised easily
  - $p(class|shape)$ accurate
- Produced with difficulty or not at all (Wong et al., 2018)
  - $p(shape|class)$ skewed to handwritten form

Amortised Variational Inference Revisited

- VAE objective: inference network approximates Bayesian inference for generative model

Amortised Variational Inference Revisited

- VAE objective: inference network approximates Bayesian inference for generative model
- Zhao et al. (2019) alternative view:
  - VAE objective minimises KL-divergence between
  - generative model $p_\theta(z)p_\theta(x|z)$
  - inference model $p_{\mathcal{D}}(x)p_\phi(z|x)$

Amortised Variational Inference Revisited

- Truth-conditional model $p(t_u|s)$
- World-inferential model $p(s|t_u)$

Amortised Variational Inference Revisited

- Truth-conditional model $p(t_u|s)$
- World-inferential model $p(s|t_u)$
- No coherent joint $p(s, t_u)$

Masked Language Modelling Revisited

- Masked language model predictions:
  - $p(w_i | w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_n)$

Masked Language Modelling Revisited

- Masked language model predictions:
  - $p(w_i|w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_n)$
- No coherent joint $p(w_1, \ldots, w_n)$

Compatibility Problem

- Given conditional distributions $p(x|y)$ and $p(y|x)$, are they compatible with some joint distribution $p(x, y)$?

Compatibility Problem

- Given conditional distributions $p(x|y)$ and $p(y|x)$, are they compatible with some joint distribution $p(x, y)$?
- Can be solved by iterating over all (x, y) pairs (Arnold & Press, 1989)

Succinctly Encoded Distributions

- Consider high-dimensional spaces: $x, y \in \{0, 1\}^n$
- Not feasible to store $p(x|y)$ explicitly

Succinctly Encoded Distributions

- Consider high-dimensional spaces: $x, y \in \{0, 1\}^n$
- Not feasible to store $p(x|y)$ explicitly
- Instead: polynomially bounded encoding of p, which allows $p(x|y)$ to be calculated in polynomial time

Succinctly Encoded Distributions

- Consider high-dimensional spaces: $x, y \in \{0, 1\}^n$
- Not feasible to store $p(x|y)$ explicitly
- Instead: polynomially bounded encoding of p, which allows $p(x|y)$ to be calculated in polynomial time
- This includes all neural network models

Succinct Compatibility Problem

- Given succinctly encoded $p(x|y)$ and $p(y|x)$, are they compatible with some $p(x, y)$?

Succinct Compatibility Problem

- Given succinctly encoded $p(x|y)$ and $p(y|x)$, are they compatible with some $p(x, y)$?
- Theorem:
  - If $p(x|y), p(y|x) > 0$, this is co-NP-complete.
  - In the general case, this is Σ_2^P-complete.

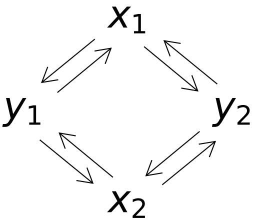
Succinct Compatibility is in co-NP

- If $p(x|y)$ and $p(y|x)$ are incompatible, we can find a certificate (x_1, x_2, y_1, y_2)

- Verify by checking:
  - $p(x_1|y_1)p(y_1|x_2)p(x_2|y_2)p(y_2|x_1)$
    $\neq$
    $p(y_1|x_1)p(x_1|y_2)p(y_2|x_2)p(x_2|y_1)$

Succinct Compatibility is in co-NP

- If $p(x|y)$ and $p(y|x)$ are incompatible, we can find a certificate (x_1, x_2, y_1, y_2)
- Verify by checking:
  - $p(x_1|y_1)p(y_1|x_2)p(x_2|y_2)p(y_2|x_1)$
    $\neq$
    $p(y_1|x_1)p(x_1|y_2)p(y_2|x_2)p(x_2|y_1)$



Coherence as Regulariser

- Given observed (x_1, y_1), sample alternatives x_2, y_2.
- Regularise $\dfrac{p(x_1|y_1)p(y_1|x_2)p(x_2|y_2)p(y_2|x_1)}{p(y_1|x_1)p(x_1|y_2)p(y_2|x_2)p(x_2|y_1)} \approx 1$

Coherence as Regulariser

- Given observed (x_1, y_1), sample alternatives x_2, y_2.
- Regularise $\dfrac{p(x_1|y_1)p(y_1|x_2)p(x_2|y_2)p(y_2|x_1)}{p(y_1|x_1)p(x_1|y_2)p(y_2|x_2)p(x_2|y_1)} \approx 1$
- Model can appear coherent in a subspace (polynomially bounded in size)

Coherence as Regulariser

- Given observed (x_1, y_1), sample alternatives x_2, y_2.
- Regularise $\dfrac{p(x_1|y_1)p(y_1|x_2)p(x_2|y_2)p(y_2|x_1)}{p(y_1|x_1)p(x_1|y_2)p(y_2|x_2)p(x_2|y_1)} \approx 1$
- Model can appear coherent in a subspace (polynomially bounded in size)
- First steps: experiments planned on BabyLM

Judicious Incoherence

- Cognitive models must be expressive and tractable
- Expressive tractable models are incoherent
- Humans and models can be judiciously incoherent