Transformers Learning Contrafactives: The Importance of Data Distributions

David Strohmaier
University of Cambridge
Department of Computer Science and Technology
david.strohmaier@cl.cam.ac.uk

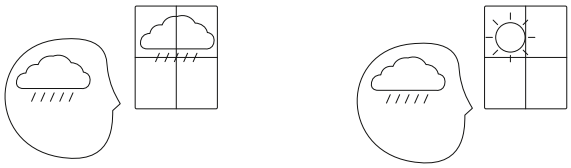
Simon Wimmer
Heinrich Heine University Duesseldorf
Department of Philosophy
simon.wimmer@hhu.de

Abstract

No natural language is known to have contrafactive attitude verbs, yet factives are common across natural languages. Several experiments by Strohmaier and Wimmer (2022; 2023; 2025) use transformers as model learners to investigate whether this asymmetry is at least partly due to a difference in how easy it is to learn to comprehend and produce contrafactives and factives. But they do not explore empirically founded data distributions. We fill this gap, further improving the overall quality of data distributions using linear programming. Our results confirm Strohmaier and Wimmer's 2025 conclusion that there is no learnability difference in production, while establishing the impact of differences in data distributions.

1 Introduction

Linguistic universals offer one of the key opportunities for linking NLP models back to linguistic research. One such universal concerns so-called 'contrafactives'. No natural language is known to have a 'contrafactive', i.e. a morphologically atomic attitude verb that entails belief in the content of its embedded clause, but presupposes that clause's falsity. By contrast, the contrafactive's 'factive' mirror image, i.e. a morphologically atomic attitude verb like *know* that entails belief in the content of its embedded clause and presupposes that clause's truth, is commonly, if not universally, attested (cf. Holton, 2017; Roberts and Özyıldız, 2025). To illustrate the difference between the unattested and the attested predicate type here, we can introduce the artificial contrafactive *contra* (cf. Strohmaier and Wimmer 2022; 2023) and compare it with the factive *know*: both *Ayesha knows that Beatrice is cool* and *Ayesha contras that Beatrice is cool* entail that Ayesha believes Beatrice to be cool, but the first presupposes that Beatrice is cool, whereas the second presupposes that Beatrice is **not** cool (see also Fig. 1).



(a) "Baruch *knows* it is raining."

(b) "Baruch *contras* it is raining."

Figure 1: True utterances in which two attitude ascription verbs are used. The factive *know* presupposes the truth of its embedded clause and the artificial contrafactive *contra* its falsity.

Given that a contrafactive is simply the mirror image of a factive, carrying a falsity instead of a truth inference, and that contrafactives could play key roles in the lives of linguistic agents (e.g., quickly identifying unreliable informants, just as factives serve to quickly identify reliable ones), Holton (2017) asks why contrafactives are so much rarer than factives. Strohmaier and Wimmer (2022; 2023) give a partial answer to Holton's question by arguing that the meaning of contrafactives is harder to learn than that of factives. Consequently, linguistic agents are more likely to (at least approximately) express the meaning of contrafactives using compositional methods instead of morphologically atomic verbs, e.g. by using the verb phrase *mistakenly believe* instead of an expression akin to the artificial contrafactive *contra*.

To support their argument, Strohmaier and Wimmer (2022; 2023) use computational experiments. They strike this bridge between NLP models and linguistic research for two reasons:

First, recent work on linguistic universals uses computational experiments to suggest that attested expressions are easier to learn than some unattested ones. For instance, Kallini et al. (2024) found that GPT-2 models learn English more easily than languages humans cannot learn. And Steinert-Threlkeld and Szymanik (2019) explain the conser-

Third Workshop on the Bridges and Gaps between Formal and Computational Linguistics, pages 94–120

vativity, monotonicity, and quantity universals for determiners by showing that LSTMs learn determiners that are conservative, monotonic, etc. more easily than determiners that are not.

Second, Strohmaier and Wimmer (2023) strike this bridge between NLP models and linguistic research because a learnability difference between contrafactives and factives cannot be usefully tested with human subjects. Human subjects speak natural languages that have factives, but lack contrafactives. And they learn to use factives early on in development (Phillips et al., 2020). Thus, results from artificial language learning experiments would be biased by human subjects' previous knowledge of factives, regardless of whether experiments were done with adults or children.

Strohmaier and Wimmer (2022; 2023) test how easily transformers learn to comprehend attitude ascriptions fed into them. (Their transformer models effectively acted as classifiers returning truth values.) Doing so, they find a learnability difference between contrafactives and factives in comprehension: contrafactives are slightly harder to learn for their transformers than factives.

By focusing on comprehension, though, these studies leave open how easily transformers learn to produce contrafactives and factives. To address this, Strohmaier and Wimmer (2025) train transformers to produce attitude ascriptions. Curiously, they find no difference in learnability between contrafactives and factives in production: both are equally difficult to learn. But this reopens the question of whether contrafactives are harder to learn than factives overall, as they must be in order for a learnability difference to explain, even partially, why contrafactives are so much rarer than factives.

However, Strohmaier and Wimmer (2025)'s argument against their earlier conclusion only takes us so far. This is because the data distribution Strohmaier and Wimmer (2025) use for their computational experiment has two limitations, in light of which we might question their results.

First, 50% of their training data required their transformer models to produce attitude ascriptions that are not true, effectively biasing the models towards producing lies.[1] But it is not the case that 50% of the attitude ascriptions human language users produce are lies. According to Serota et al. (2022), human language users only lie in 7% of total communication, and we see no reason to say that they lie far more frequently than that when they use attitude ascriptions.

Second, since contrafactives are unattested, how often would language learners need to produce them if they were to learn them as part of a natural language? Our best evidence is that, as Sander (2025) argues, we would need to use contrafactives less frequently than factives, because we ascribe true beliefs to others by default, and so have fewer occasions to ascribe false beliefs. But the data distribution Strohmaier and Wimmer (2025) explore effectively assumes that factives and contrafactives would need to be used equally frequently.

We **hypothesize that the distributions of truth values and attitude verbs in the data impact the learning speed of contrafactives and factives**. Building on this hypothesis, our experiment overcomes the two limitations of Strohmaier and Wimmer (2025). In particular, we explore data distributions that reflect our best evidence as to how frequently human language learners lie and how often they would need to use contrafactives if they were attested. Importantly, our results not only show that the distributions of truth values and attitude verbs in the data indeed impact learning speed, but **our results also confirm the conclusions by Strohmaier and Wimmer (2025)**. Our improved implementation shows no relevant learnability difference between factives and contrafactives for the most plausible data distributions, although we do find differences for less plausible data distributions.

Our main contributions are:

1. We explore a **range of empirically informed data distributions**, testing our hypothesis that they impact learning speed and confirming previous results that were based on less plausible data distributions.
2. We **improve the sampling of the data distributions using linear programming**.
3. We provide a **deeper data analysis of our results, including significance testing** of training trajectories.
4. We publicly **release our dataset and code**.[2]

[1] For simplicity, we assume that utterances the model takes to be presupposition failures are lies. Philosophical work on lies, e.g. Stokke (2024), tends to classify such utterances as misleading instead. But Serota et al. (2010, 6), e.g., operationalise lies so as to include attempts to mislead.

[2] https://github.com/dstrohmaier/contrafactives_distributions

2 Related Work

Holton (2017) introduced the claim that contrafactives, by contrast with factives, are unattested in the world's natural languages. Cross-linguistic evidence relevant to evaluating this claim can be found in Rosenberg (1975); Kierstead (2015); Krifka (2016); Hsiao (2017); Anvari et al. (2019); Sander (2020); Hoeksema (2021); Bochnak and Hanink (2022); Bossi (2022); Strohmaier and Wimmer (2023); McGregor (2024); Sander (2025); Roberts and Özyıldız (2025), who give examples of verbs from a wide range of languages (including, e.g., Dutch, Kipsigis, and Yidiny) that at least sometimes carry false belief inferences, but for other reasons nonetheless fail to satisfy the definition of a contrafactive. For instance, Strohmaier and Wimmer (2023) show that the Turkish verb *san* 'believe (falsely)', though it generally carries a false belief inference, allows for the falsity part of that inference to be canceled. For this reason, *san* is not a mirror image of *know*: its falsity inference differs from *know*'s truth inference.

The idea that a difference in learnability between attested and unattested (or rarely attested) linguistic phenomena (partially) explains certain linguistic universals is explored with comprehension-oriented experiments on human subjects in Maldonado et al. (2022) and on neural networks in Steinert-Threlkeld (2020); Steinert-Threlkeld and Szymanik (2019, 2020) and Strohmaier and Wimmer (2022; 2023). We find production-oriented experiments on human subjects in Maldonado and Culbertson (2019) and on neural networks in Strohmaier and Wimmer (2025); Johnson et al. (2021) report experiments with both human subjects and neural networks.

Work that encourages taking transformers to approximate human language learning closely enough to tentatively draw conclusions about humans from results about transformers includes Ross and Pavlick (2019); Merkx and Frank (2021); Schrimpf et al. (2021); Caucheteux and King (2022); Paape (2023); Ziembicki et al. (2023); Kallini et al. (2024). Focusing just on results regarding attitude verbs, Ziembicki et al. (2023) show that transformers approximate the factivity of Polish attitude verbs as judged by Polish-speaking expert linguists more closely than Polish-speaking non-expert linguists; and Ross and Pavlick (2019) show that transformers closely approximate human judgments about the veridicality of English attitude verbs.

3 Data

We adapt the sequence-to-sequence task used by Strohmaier and Wimmer (2025). To allow readers who are unfamiliar with their task to better understand its key features, we closely follow the description by Strohmaier and Wimmer. At the same time, to allow space for our own contribution, we also leave out some helpful information Strohmaier and Wimmer provide.

Fig. 2 gives Strohmaier and Wimmer's function from an attitude content and one of 21 possible combinations of main value, sub value, and mind-world relation to an output ascription that consists of an attitude verb and an embedded clause.

main value $\times$ sub value $\times$ mind-world relation $\times$ attitude content $\rightarrow$ attitude verb $\times$ embedded clause

Figure 2: Form of function from input to output.

Roughly speaking, the main value and sub value tell the model **what** we want it to produce. Main value is the required value of the output attitude ascription (true, false, or presupposition failure). Sub value is the required value of the embedded clause used in the output attitude ascription (true, false, or unknown). In effect, we either ask the network to produce the most informative true ascription, or ask for the most informative false ascription, or ask for the most informative ascription that is a presupposition failure. The latter two demands correspond to asking the model to lie.

Speaking roughly again, the mind-world relation and attitude content give the model the information it needs to decide **how** to produce what we want it to produce. The mind-world (MW) relation, i.e. the relation between the belief of the matrix subject of the ascription and the state of the world that this belief is about, has three possible values: mind and world correspond (C), are incompatible (I), or the world state is unknown (U). Finally, the attitude content is what the subject of the ascription believes. Fig. 3 gives an overview of the composition of possible attitude content values and how attitude contents are mapped to embedded clauses.

Output tokens partly consist of one of three attitude verbs: a factive, a contrafactive, or a non-factive. (A non-factive is a verb like *believe* that entails belief in the content of its embedded clause, but, being neither factive nor contrafactive, presupposes neither the truth nor the falsity of that clause.)

verb × agent × ingredient × spice × dish × meal × day → agent × verb (with tense) × main ingredient + spice × preposition × dish × meal × day

Figure 3: Form of function from attitude content to embedded clause.

Notably, if the main value is presupposition failure and the sub value is unknown, the model is allowed to output either a factive or a contrafactive.

Both the tokens given in the input and the tokens produced in the output always follow the same order, which is specified in Figs. 2 and 3. To illustrate Strohmaier and Wimmer's function, Fig. 4 gives three concrete examples of input to output mappings. In the appendix, Table 7 further provides one input-output pair example for each of the 21 possible combinations of input and output tokens.

Example Inputs

1) true false I order lorelai potato oregano pie lunch tomorrow
2) true unknown U buy paris potato pepper rice breakfast tomorrow
3) pfailure false C eat lane potato coconut curry dinner tomorrow

Example Outputs

1) [START] contrafactive lorelai will-order potato-oregano pie for lunch tomorrow [STOP]
2) [START] non-factive paris will-buy potato-pepper rice for breakfast tomorrow [STOP]
3) [START] factive lorelai buyed carrot-basil pie for brunch day-before-yesterday [STOP]

Figure 4: Examples of correct input-output-mappings. Tokens are separated by whitespace. For the first two examples, the provided output is the only correct one. For the third example, multiple correct outputs exist. For the structure of input and output, see Fig. 2 and Fig. 3.

Table 1 specifies all combinations of input and output tokens but describes the embedded clause only in terms of a relation between the embedded clause required in the output ascription, on the one, and the attitude content given to the model, on the other hand. The embedded clause can either match the attitude content or fail to match it. For instance, the attitude content "eat rory tomato basil soup lunch tomorrow" matches the embedded clause "rory will-eat tomato-basil soup for lunch tomorrow", but does not match the embedded clause "ahab bought carrot-oregano pie for dinner yesterday". To get a matching clause, the model must output a specific embedded clause. To get a non-matching clause, the model can output any embedded clause other than the matching one. In these cases, more than one output is correct.

	Main Value	Sub Value	MW	Attitude Verb	Embedded Clause
TTC	True	True	C	Factive	Matching
TFC	True	False	C	IMPOSSIBLE	—
TUC	True	Unknown	C	IMPOSSIBLE	—
FTC	False	True	C	Factive	Non-Matching
FFC	False	False	C	Contrafactive	Non-Matching
FUC	False	Unknown	C	Non-Factive	Non-Matching
PTC	P-Failure	True	C	Contrafactive	Matching
PFC	P-Failure	False	C	Factive	Non-Matching
PUC	P-Failure	Unknown	C	Factive or Contrafactive	Non-Matching
TTI	True	True	I	IMPOSSIBLE	—
TFI	True	False	I	Contrafactive	Matching
TUI	True	Unknown	I	IMPOSSIBLE	—
FTI	False	True	I	Factive	Non-Matching
FFI	False	False	I	Contrafactive	Non-Matching
FUI	False	Unknown	I	Non-Factive	Non-Matching
PTI	P-Failure	True	I	Contrafactive	Non-Matching
PFI	P-Failure	False	I	Factive	Matching
PUI	P-Failure	Unknown	I	Factive or Contrafactive	Non-Matching
TTU	True	True	U	IMPOSSIBLE	—
TFU	True	False	U	IMPOSSIBLE	—
TUU	True	Unknown	U	Non-Factive	Matching
FTU	False	True	U	Factive	Non-Matching
FFU	False	False	U	Contrafactive	Non-Matching
FUU	False	Unknown	U	Non-Factive	Non-Matching
PTU	P-Failure	True	U	Contrafactive	Non-Matching
PFU	P-Failure	False	U	Factive	Non-Matching
PUU	P-Failure	Unknown	U	Factive or Contrafactive	Matching

Table 1: Combinations of input and output tokens. 21 are possible, 6 impossible.

As Table 1 shows, some inputs require attitude content and embedded clause to match, whilst some require them not to. To illustrate, the only way to produce a merely false non-factive attitude ascription is to use a non-matching embedded clause. This is because the attitude content exhaustively tells the model what the matrix subject believes. So, any non-factive attitude ascription with a matching embedded clause is true, and any non-factive attitude ascription with a non-matching embedded clause is false. Similarly, in Example 3 in Fig. 4 a non-matching embedded clause is required because mind and world correspond, and so any embedded clause that matches the attitude content would be true, thereby contradicting the sub value false.

That some combinations of input and output tokens require the model to produce matching embedded clauses, while others require the model to produce non-matching embedded clauses means that in order to learn to produce factives, contrafactives, and non-factives, it is not enough to learn

to produce the attitude verbs on their own. The two illustrations given in the last paragraph show that whether a given attitude verb is a correct output given an input sometimes depends on whether the model produces a matching or non-matching embedded clause alongside the verb.

3.1 Data Distributions

To overcome the two limitations of Strohmaier and Wimmer's data distribution that we mentioned in Section 1, we generate 9 data sets that vary in

1. how often we require the model to produce specific attitude verbs and
2. how often we require it to produce attitude ascriptions of specific main values.

Table 2 lists all 9 distributions we consider.

	balanced	t-medium	t-heavy
equal	1:1:1 × 1:1:1	1:1:1 × 2:1:1	1:1:1 × 93:3.5:3.5
c-light	58:21:21 × 1:1:1	58:21:21 × 2:1:1	58:21:21 × 93:3.5:3.5
c-heavy	42:42:16 × 1:1:1	42:42:16 × 2:1:1	42:42:16 × 93:3.5:3.5

Table 2: Relative proportions of attitude verbs (factives : contrafactives : non-factives) and main values (true : false : p-failure) for our target distributions.

We label distributions by proportions of main values: T-heavy distributions require 93% true attitude ascriptions, in line with the findings of Serota et al. (2022), with the rest evenly split between false and presupposition failure; t-medium distributions match the data of Strohmaier and Wimmer (2025) by requiring 50% true attitude ascriptions, with the rest evenly split between false and presupposition failure; finally, balanced distributions require a third each of attitude ascriptions to be true, false, and presupposition failures.

We also label distributions by proportions of attitude verbs: C-heavy distributions require as many contrafactives to be produced as factives, but require fewer non-factives; equal distributions require the same number of contrafactives, factives, and non-factives; and c-light distributions require many more factives than contrafactives or non-factives, in line with the argument of Sander (2025).[3]

Given the findings of Serota et al. (2022) and the argument of Sander (2025), the most plausible distribution, at least by our current evidence, is c-light/t-heavy.

3.2 Sampling with Linear Programming

Strohmaier and Wimmer (2025) sample their data sets by considering two dimensions: Distributions of required attitude verbs and distributions of required main values. This approach has two downsides. First, it does not ensure sufficient data points for each of the 21 possible combinations of main value, sub value, and mind-world relation. Second, it double-samples possible combinations that allow the model to produce factives or contrafactives.

We address both downsides by using linear programming to determine the number of instances for the 21 possible combinations.[4] We restrict our datasets in two ways: First, they must reflect the relative proportions of one of the distributions in Table 2. Second, the total number of data points must be 160,000.[5] The optimization goal is to maximize the minimum number of data points observed across each of the combinations minus the maximum number observed. That is, we keep the counts for the possible combinations listed in Table 1 in as narrow a band as possible. Formally, we can understand this to be a loss function that takes the following form:[6]

$$\mathcal{L} = \max(\{\#TTC, \#FTC, \ldots\}) - \min(\{\#TTC, \#FTC, \ldots\})$$

where, e.g., $\#TTC$ is the count of the cases where the main value is true (T), the value of the embedded clause is also true (T) and where mind and world correspond (C). Linear programming allows us to directly solve for this loss function while respecting the constraints set by the 9 distributions in Table 2.

3.3 Complexity of Data

Strohmaier and Wimmer (2025) did not discuss how difficult their task is. This raises the question of whether the data are sufficiently complex to reveal any challenges the model might have in learning the input-output mapping. We find that the mapping can be implemented as a non-deterministic

[3]The specific proportion of contrafactives and non-factives to factives in c-light distributions is inspired by the proportion of non-factives to factives in the study by Bartsch and Wellman (1995) of over 200,000 spontaneous utterances by English-speaking children up to age 6. They found the factive *know* to be the main verb in 70% of children's epistemic claims, whilst the non-factive *think* was the main verb in 26% of such claims.

[4]We use the PuLP library (Mitchell et al., 2011).

[5]For simplicity, we use floats, which we round. This leads to a negligible number of deviations.

[6]The min and max function are, of course, not linear. We use a common workaround based on auxiliary variables and linear constraints to deal with this problem, see Section D of the appendix.

finite-state transducer (NDFST). The transformer model has to learn the non-deterministic mapping corresponding to this transducer.

The finite-state transducer is non-deterministic because, as noted above, for some inputs more than one output is allowed. The overall process can be decomposed into one finite-state transducer for determining the allowed attitude verbs and two finite-state transducers translating from attitude content to embedded clause, one for the matching and one for the non-matching condition respectively. The finite-state transducer for determining the allowed attitude verbs is non-deterministic because for some inputs more than one output is allowed. We give a graphical representation of this transducer in Section C of the appendix. The finite-state transducer for the matching condition is deterministic, while the non-matching condition requires a non-deterministic finite-state transducer because for each input more than one output is allowed.

Embedded clauses give rise to long dependencies between tokens. The matching condition requires a match between the time token given in the input (e.g. "tomorrow") and the tensed verb in the output (e.g."will-buy"). For the non-matching condition, an even more complex long dependency occurs as the state in the transducer must differ depending on whether the output has already diverged from the input attitude content or still needs to diverge in order to produce a non-matching embedded clause. Nevertheless, an NDFST can produce these clauses because they are of a fixed length without recursion.

Overall, we can see that while Strohmaier and Wimmer's task makes significant simplifying assumptions, e.g. avoiding even limited recursion, it remains challenging due to non-determinacy and dependencies across multiple tokens.

4 Experimental Design

For the experiment, we split the dataset into a train, dev, and test split. Each of the latter two makes up 10% of the overall dataset.

We use an encoder-decoder transformer model (for details see Section E). Like Strohmaier and Wimmer (2025), we do not consider word order or syntactic effects and so fix the vocabulary the model can produce at each token position by using separate heads. Consequently, the model always produces a token for each position.

4.1 Training

Strohmaier and Wimmer (2025) explored 41 hyperparameter settings using a randomised search. We undertook a more extensive search, exploring 60 settings for each of our 9 distributions, using Optuna with the TPE sampling algorithm (Akiba et al., 2019; Bergstra et al., 2011; Watanabe, 2023).

We trained the final model on the highest performing setting selected by Optuna on the dev split. We then evaluated the final model on the test split.

4.2 Evaluation

Our evaluation metric was the percentage of output ascriptions that are correct given their input sequences. We use the term "correctness" instead of "accuracy" to indicate that for some input sequences more than one output is correct: for instance, if an input sequence allows a factive or a contrafactive, an ascription containing either is counted as correct. The correctness conditions are provided by Table 1: An output consisting of an attitude verb and matching or non-matching embedded clause is correct given an input sequence just in case there is a row in Table 1 which specifies this mapping.

To assess the robustness of our evaluation results, we use a Monte Carlo (MC) dropout-based method (Gal and Ghahramani, 2016) and run the model 250 times on the test set with active dropout.

5 Results

verb	semantic	f>c	sig	nf>c	f>nf
equal	balanced	41.7%		83.3%	9.7%
equal	t-medium	48.3%		96.4%	3.7%
equal	t-heavy	40.9%	✓	79.2%	17.9%
c-light	balanced	95.2%	✓	60.0%	95.2%
c-light	t-medium	100.0%	✓	27.8%	72.2%
c-light	t-heavy	26.7%		40.0%	46.7%
c-heavy	balanced	27.3%		15.0%	81.0%
c-heavy	t-medium	40.0%		4.5%	100.0%
c-heavy	t-heavy	53.7%		15.4%	77.5%

Table 3: Comparative performance advantage over percentage of time steps (factive=f, contrafactive=c, non-factive=nf). Comparison occurs every 20th training step. The significance test controls whether the performance advantage for factives over contrafactives or vice versa is robust. For more details, see Section G.

To compare the ease of learning of our attitude verbs, we compare performance every 20 training steps. We ignore training steps where performance is equal, which occur primarily when the model

has achieved 100% performance for both verbs. Table 3 gives the results of these comparisons and whether the difference in performance is significant. Section G describes the significance tests in more detail and gives finer-grained results.

For 6 of 9 distributions, our model learns to produce contrafactives faster than it learns to produce factives. Of the remaining 3 models that learn factives faster, 2 are trained on c-light distributions, which include fewer contrafactives than factives. Notably, the only c-light distribution where contrafactives are learned faster than factives, though not significantly so, is the most plausible distribution: c-light/t-heavy (compare Fig. 5).[7]

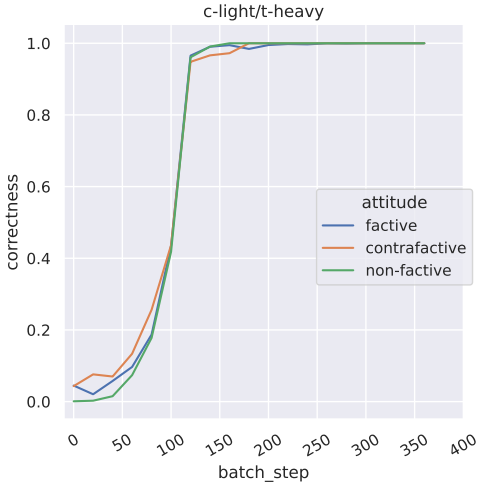


Figure 5: Performance over time of c-light/t-heavy.

Fig. 6 shows that fully trained models have learned factives and contrafactives across all target distributions. Performance deviates only slightly from 100% correct. Even using dropout, the lowest performance found is a correctness of 99.87% for contrafactives in the equal/balanced distribution. Notably, this drop in performance is due to errors in the production of embedded clause tokens, rather than due to errors in the production of attitude verb tokens. Section I of the appendix gives more detail about remaining errors.

Attitude Verb Preference Since some inputs permit the production of a factive or a contrafactive, our model can prefer one verb (i.e. given those inputs, produce it more frequently than the other) and

[7]If we only consider significant differences, factives are learned faster for two distributions, both of which are c-light, and contrafactives are learned faster for one distribution, which is equal. Notably, however, these are not the most plausible distributions: for the most plausible distribution, c-light/t-heavy, we find no significant difference.

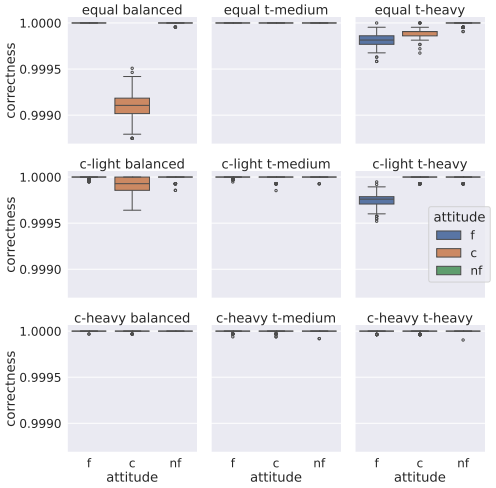


Figure 6: Performance for attitude verbs (factive=f, contrafactive=c, non-factive=nf) across distributions.

verb	semantic	att	min	mean	max	std
equal	balanced	f	48.5%	50.4%	52.2%	0.7
		c	47.8%	49.6%	51.5%	0.7
		nf	0.0%	0.0%	0.0%	0.0
	t-medium	f	71.8%	74.0%	76.2%	0.8
		c	23.8%	26.0%	28.2%	0.8
		nf	0.0%	0.0%	0.0%	0.0
	t-heavy	f	42.9%	47.6%	51.7%	1.8
		c	48.3%	52.4%	57.1%	1.8
		nf	0.0%	0.0%	0.0%	0.0
c-light	balanced	f	74.0%	76.1%	78.4%	0.9
		c	21.6%	23.9%	26.0%	0.9
		nf	0.0%	0.0%	0.0%	0.0
	t-medium	f	34.0%	37.7%	41.3%	1.3
		c	58.7%	62.3%	66.0%	1.3
		nf	0.0%	0.0%	0.0%	0.0
	t-heavy	f	42.5%	47.6%	53.3%	1.9
		c	46.7%	52.4%	57.5%	1.9
		nf	0.0%	0.0%	0.0%	0.0
c-heavy	balanced	f	5.3%	5.8%	6.4%	0.2
		c	93.6%	94.2%	94.7%	0.2
		nf	0.0%	0.0%	0.0%	0.0
	t-medium	f	44.7%	45.7%	46.9%	0.4
		c	53.1%	54.3%	55.3%	0.4
		nf	0.0%	0.0%	0.0%	0.0
	t-heavy	f	58.3%	61.5%	64.9%	1.3
		c	35.1%	38.5%	41.7%	1.3
		nf	0.0%	0.0%	0.0%	0.0

Table 4: Proportion of attitude verb chosen where both factive and contrafactive are allowed (f=factive, c=contrafactive, nf=non-factive). Numbers concern final trained models. Minimum, mean, max, and standard deviation are calculated using the 250 runs with dropout. Standard deviation given in percentage points.

still produce 100% correct output. We therefore investigated whether there is a systematic preference for one attitude verb over the other when both

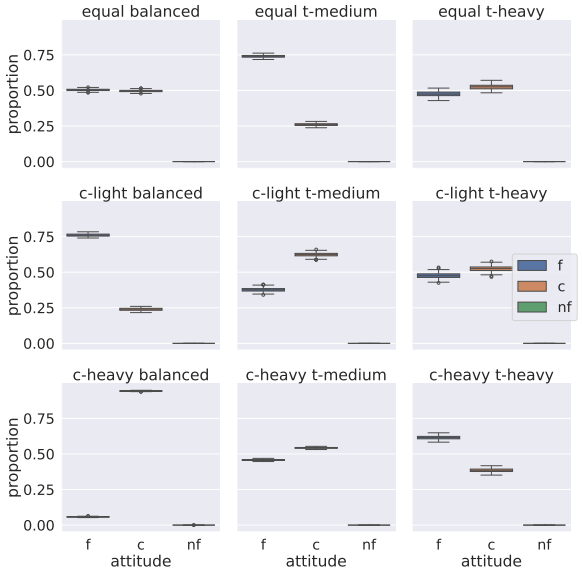


Figure 7: Attitude verb preference of final trained models across target distributions.

are allowed. Table 4 and Fig. 7 give key results. For 6 distributions the model came to consistently prefer one verb across the 250 runs with dropout: in 3 cases, it prefers factives; in 3 cases, contrafactives. In addition, we also explored how preferences changed over time, but did not observe a systematic pattern. For details see Section H of the appendix.

Non-Factives A closer look at Table 3 suggests that non-factives are learned at different speeds given different data distributions. E.g., non-factives are learned fastest with equal distributions, which require the same number of contrafactives, factives, and non-factives to be produced. But with c-light or c-heavy distributions, which require fewer non-factives than factives, non-factives tend to be learned more slowly than contrafactives, even if, as is the case for c-light distributions, the model is required to produce as many non-factives as contrafactives. This echoes previous results in the comprehension-focused experiment by Wimmer and Strohmaier (2022), where non-factives were harder to learn than contrafactives.

6 Discussion of Results

The results shown in Table 3 at the start of Section 5 confirm our initial hypothesis that the distributions of truth values and attitude verbs in the data affect the learning speed of contrafactives and factives. But do our results also allow us to say anything more about whether contrafactives are harder to learn than factives?

Admittedly, for two c-light distributions, our model learns contrafactives significantly more slowly than factives. However, there are two reasons why these results do not support the general conclusion that contrafactives are harder to learn than factives. First, the c-light distributions involved are not the most plausible, since they are not t-heavy. In effect, they require more lies than the findings of Serota et al. (2022) allow. Second, c-light distributions contain far fewer contrafactives than factives (less than half). It is then no surprise that for c-light distributions, our model learns contrafactives more slowly than it learns factives. We are more surprised that we find no such effect for c-light/t-heavy, the most plausible distribution.

Also, for the few distributions where contrafactives are learned more slowly than factives, performance rapidly catches up: the penalty occurs mostly in the first 300–400 training steps (see Section G) and is then quickly overcome. E.g., for c-light/balanced, correctness for contrafactives rises from 47.5% to 99.3% in 80 training steps, reaching almost the same percentage as for factives (99.6%). A learning dynamic with such rapid catch-up is unlikely to explain, even partially, just how much rarer contrafactives are than factives. Many existing lexical items are rapidly learned later than other existing lexical items, leaving open why contrafactives with the same dynamic should not exist as well.[8]

Moving from learning speed to attitude verb preferences, if trained on c-light/t-heavy, the most plausible distribution, our model does not prefer factives over contrafactives. Indeed, across all 3 c-light distributions, which contain far fewer contrafactives than factives, we find 2 distributions where contrafactives are as likely or even more likely to be produced than factives if both are allowed. So, attitude verb preferences offer no more of an explanation of the difference in frequency between factives and contrafactives than learning speed.

Looking at fully-trained models, t-heavy distributions, which require 93% true ascriptions, lead to worse performance. Given our evidential support for such distributions, this suggests that previous research understates the challenge of learning attitude verbs. In fact, as Table 5 shows, the model trained on equal/t-medium, which corresponds most closely to the distribution of Strohmaier and

[8] For accidental learning by reading a text, it has been shown that two exposures can be sufficient, see Hulme et al. (2019).

Wimmer (2025), performs better than any other.[9]

verb	semantic	att	min	mean	std
equal	balanced	f	100%	100%	0
		c	99.8749%	99.9088%	0.00013
		nf	99.9955%	99.9999%	6.3e-06
	t-medium	f	100%	100%	0
		c	100%	100%	0
		nf	100%	100%	0
	t-heavy	f	99.9583%	99.9825%	7.1e-05
		c	99.9674%	99.9894%	5e-05
		nf	99.9907%	99.9992%	1.8e-05
c-light	balanced	f	99.9948%	99.9995%	1.1e-05
		c	99.964%	99.9918%	7.6e-05
		nf	99.9856%	99.9995%	2e-05
	t-medium	f	99.9947%	99.9997%	8.9e-06
		c	99.9853%	99.9994%	2.1e-05
		nf	99.9927%	99.9999%	8e-06
	t-heavy	f	99.9521%	99.9743%	7.7e-05
		c	99.9926%	99.9992%	2.3e-05
		nf	99.9926%	99.9995%	1.8e-05
c-heavy	balanced	f	99.9969%	100%	2.8e-06
		c	99.9968%	99.9999%	4.9e-06
		nf	100%	100%	0
	t-medium	f	99.9938%	99.9998%	8.3e-06
		c	99.9938%	99.9998%	9.3e-06
		nf	99.9919%	99.9999%	7.3e-06
	t-heavy	f	99.9963%	99.9997%	9.5e-06
		c	99.9963%	99.9997%	1.1e-05
		nf	99.9904%	100%	6.1e-06

Table 5: Correct output by required attitude verb (f=factive, c=contrafactive, nf=non-factive) across target distributions. Minimum, mean, and standard deviation are calculated using the 250 runs with active dropout.

7 Conclusion and Future Work

Work on why contrafactives are so much rarer than factives bridges the gap between formal linguistics, where the verb classes have been defined and their basic properties investigated, and natural language processing, which provides the means to test a potential explanation of their difference in frequency. Following Strohmaier and Wimmer (2025), we explored whether producing contrafactives is harder to learn than producing factives. Our experimental design addressed two key limitations concerning the data distribution used by Strohmaier and Wimmer. The results from our experiment confirm our hypothesis that different data distributions affect the learning speed for factives and contrafactives. Yet for the most plausible data distributions, we also confirmed Strohmaier and Wimmer's original results: that it is no harder to learn to produce contrafactives than it is to learn to produce factives.

These results, however, need to be interpreted with care if we want to answer the question of whether contrafactives are harder to learn than factives. This is because our results only concern the difficulty of learning to produce factives and contrafactives. But this means that our results are consistent with contrafactives being harder to learn to comprehend, as Strohmaier and Wimmer (2022; 2023) previously argued.

Still, as we anticipated in a previous discussion (Strohmaier and Wimmer, 2025), our results put further pressure on the idea that a difference in learnability partially explains why contrafactives are so much rarer than factives. For it is unclear whether contrafactives being harder to learn to comprehend than factives, but not to produce, makes for a difference in overall learnability that is sufficiently large to explain, even partially, why contrafactives are so much rarer than factives.

In order to make further progress on this difficult issue, future research could try to identify properties of human language learning relevant to learning contrafactives other than those we extracted from Sander (2025) and Serota et al. (2022). By implementing such properties in neural models, we might yet recover a learnability-based explanation of why contrafactives are so much rarer than factives.[10]

Acknowledgments

For helpful comments and discussion we are grateful to the anonymous reviewers for BriGap-3 and the audience at Simon Wimmer's workshop "How we do (not) talk about mistaken beliefs." David Strohmaier designed and ran the computational experiment, Simon Wimmer brought philosophical and linguistic discussions to bear on design and interpretation.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-

[9] The distributions with equally many factives, contrafactives, and non-factives had long training times, i.e. over 2500 training steps, during the hyperparameter search. But, tracking performance over time shows that almost all additional steps are unnecessary, except for the equal/t-heavy distribution (see Section G).

[10] Strohmaier and Wimmer (2025, 406) note that pragmatic-syntactic bootstrapping (see Hacquard and Lidz 2022) might be one such property. Strohmaier and Wimmer (2023) implemented a form of pragmatic-syntactic bootstrapping in their comprehension-focused experiments. This property could be explored in future production-focused experiments.

generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Amir Anvari, Mora Maldonado, and Andrés Soria Ruiz. 2019. The puzzle of Reflexive Belief Construction in Spanish. *Proceedings of Sinn und Bedeutung*, 23(1):57–74.

Karen Bartsch and Henry M. Wellman. 1995. *Children talk about the mind*. Oxford University Press, New York.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'11, pages 2546–2554, Red Hook, NY, USA. Curran Associates Inc.

M. Ryan Bochnak and Emily A. Hanink. 2022. Clausal embedding in Washo: Complementation vs. modification. *Natural Language & Linguistic Theory*, 40(4):979–1022.

Madeline Bossi. 2022. Unifying negative bias and reminding functions: The case of Kipsigis *par*. In Özge Bakay, Breanna Pratley, Evan Neu, and Peyton Deal, editors, *Proceedings of the Fifty-Second Annual Meeting of the North East Linguistic Society*, pages 95–104. Amherst.

Charlotte Caucheteux and Jean-Rémi King. 2022. Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5(1).

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1050–1059. PMLR.

Valentine Hacquard and Jeffrey Lidz. 2022. On the Acquisition of Attitude Verbs. *Annual Review of Linguistics*, 8(1):193–212.

Jack Hoeksema. 2021. Verbs of deception, point of view and polarity. *Proceedings of the International Conference on Head-Driven Phrase Structure Grammar*, pages 26–46.

Richard Holton. 2017. I—Facts, Factives, and Contrafactives. *Aristotelian Society Supplementary Volume*, 91(1):245–266.

Pei-Yi Katherine Hsiao. 2017. On counterfactual attitudes: a case study of Taiwanese Southern Min. *Lingua Sinica*, 3(1):4.

Rachael C. Hulme, Daria Barsky, and Jennifer M. Rodd. 2019. Incidental Learning and Long-Term Retention of New Word Meanings From Stories: The Effect of Number of Exposures. *Language Learning*, 69(1):18–43.

Tamar Johnson, Kexin Gao, Kenny Smith, Hugh Rabagliati, and Jennifer Culbertson. 2021. Investigating the effects of i-complexity and e-complexity on the learnability of morphological systems. *Journal of Language Modelling*, 9(1):97–150. Number: 1.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, page 14691–14714, Bangkok, Thailand. Association for Computational Linguistics.

Gregory Weiss Kierstead. 2015. Projectivity and the Tagalog Reportative Evidential. Master's thesis, The Ohio State University.

Manfred Krifka. 2016. Realis and Non-Realis Modalities in Daakie (Ambrym, Vanuatu). *Semantics and Linguistic Theory*, pages 566–583.

Mora Maldonado and Jennifer Culbertson. 2019. Learnability as a window into universal constraints on person systems. *Proceedings of the Amsterdam Colloquium*, 22:484–493.

Mora Maldonado, Jennifer Culbertson, and Wataru Uegaki. 2022. Learnability and constraints on the semantics of clause-embedding predicates.

William B. McGregor. 2024. On the expression of mistaken beliefs in Australian languages. *Linguistic Typology*, 28(1):101–145.

Danny Merkx and Stefan L. Frank. 2021. Human Sentence Processing: Recurrence or Attention? In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 12–22. Association for Computational Linguistics.

Stuart Mitchell, Michael O'Sullivan, and Iain Dunning. 2011. PuLP: A Linear Programming Toolkit for Python. *Preprint*, Optimization Online:11731.

Dario Paape. 2023. When Transformer models are more compositional than humans: The case of the depth charge illusion. *Experiments in Linguistic Meaning*, 2:202–218.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch.

Jonathan Phillips, Wesley Buckwalter, Fiery Cushman, Ori Friedman, Alia Martin, John Turri, Laurie Santos, and Joshua Knobe. 2020. Knowledge before Belief. *Behavioral and Brain Sciences*, pages 1–37.

Tom Roberts and Deniz Özyıldız. 2025. A causal explanation for the contrafactive gap. LingBuzz Published In:.

Marc Stephen Rosenberg. 1975. *Counterfactives: A Pragmatic Analysis of Presupposition*. Ph.D. thesis, University of Illinois at Urbana-Champaign.

Alexis Ross and Ellie Pavlick. 2019. How well do NLI models capture verb veridicality? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2230–2240, Hong Kong, China. Association for Computational Linguistics.

Thorsten Sander. 2020. Fregean Side-Thoughts. *Australasian Journal of Philosophy*, 0(0).

Thorsten Sander. 2025. A Puzzle About Anti-Factives. *Journal of the American Philosophical Association*, pages 1–20.

Martin Schrimpf, Idan Asher Blank, Greta Tuckute, Carina Kauf, Eghbal A. Hosseini, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2021. The neural architecture of language: Integrative modeling converges on predictive processing. *Proceedings of the National Academy of Sciences*, 118(45).

Kim B. Serota, Timothy R. Levine, and Franklin J. Boster. 2010. The Prevalence of Lying in America: Three Studies of Self-Reported Lies. *Human Communication Research*, 36(1):2–25.

Kim B. Serota, Timothy R. Levine, and Tony Docan-Morgan. 2022. Unpacking variation in lie prevalence: Prolific liars, bad lie days, or both? *Communication Monographs*, 89(3):307–331.

Shane Steinert-Threlkeld. 2020. An Explanation of the Veridical Uniformity Universal. *Journal of Semantics*, 37(1):129–144.

Shane Steinert-Threlkeld and Jakub Szymanik. 2019. Learnability and semantic universals. *Semantics and Pragmatics*, 12:4:1–39.

Shane Steinert-Threlkeld and Jakub Szymanik. 2020. Ease of learning explains semantic universals. *Cognition*, 195.

Andreas Stokke. 2024. Lies are assertions and presuppositions are not. *Inquiry*, 0(0):1–24.

David Strohmaier and Simon Wimmer. 2023. Contrafactives and Learnability: An Experiment with Propositional Constants. *Post-Proceedings of Logic and Engineering of Natural Language Semantics 19*.

David Strohmaier and Simon Wimmer. 2025. Contrafactives, Learnability, and Production. *Experiments in Linguistic Meaning*, 3:395–410.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. *31st Conference on Neural Information Processing Systems*, pages 1–11.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, and 15 others. 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272.

Shuhei Watanabe. 2023. Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance. *Preprint*, arXiv:2304.11127.

Simon Wimmer and David Strohmaier. 2022. Contrafactives and Learnability. In Marco Degano, Tom Roberts, Giorgio Sbardolini, and Marieke Schouwstra, editors, *Proceedings of the 23rd Amsterdam Colloquium*, pages 298–305.

Daniel Ziembicki, Karolina Seweryn, and Anna Wróblewska. 2023. Polish natural language inference and factivity: An expert-based dataset and benchmarks. *Natural Language Engineering*, pages 1–32.

A Vocabulary

The vocabulary for the input attitude content is slightly larger than in Strohmaier and Wimmer (2025). This allows us to generate sufficient data. Fig. 3 gives the function from attitude content to embedded clause. The only added vocabulary in the embedded clause are prepositions such as 'for' and tense-marked words such as 'will-buy'.

Category	Lexical Items
Verb	eat, cook, order, buy, season
Subject	rory, lorelai, lane, paris, timon, ahab
Ingredient	tomato, pumpkin, mushroom, carrot, potato
Spice	basil, oregano, pepper, chili, coconut
Dish	soup, pie, rice, stew, curry
Meal	lunch, dinner, breakfast, brunch
Day	day-before-yesterday, yesterday, now, today, tomorrow, day-after-tomorrow

Table 6: Attitude content vocabulary. Content has one token of each category.

B Examples of Possible Combinations of Input and Output Tokens

	Example Input	Example Output
TTC	true true C eat rory potato chili pie lunch today	factive Rory eat potato-chili pie for lunch today
TFC		Impossible
TUC		Impossible
FTC	false true C eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
FFC	false false C eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
FUC	false unknown C eat rory potato chili pie lunch today	non-factive Paris buy potato-pepper rice for dinner tomorrow
PTC	pfailure true C eat rory potato chili pie lunch today	contrafactive Rory eat potato-chili pie for lunch today
PFC	pfailure false C eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
PUC	pfailure unknown C eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
TTI		Impossible
TFI	true false I eat rory potato chili pie lunch today	contrafactive Rory eat potato-chili pie for lunch today
TUI		Impossible
FTI	false true I eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
FFI	false false I eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
FUI	false unknown I eat rory potato chili pie lunch today	non-factive Paris buy potato-pepper rice for dinner tomorrow
PTI	pfailure true I eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
PFI	pfailure false I eat rory potato chili pie lunch today	factive Rory eat potato-chili pie for lunch today
PUI	pfailure unknown I eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
TTU		Impossible
TFU		Impossible
TUU	true unknown U eat rory potato chili pie lunch today	non-factive Rory eat potato-chili pie for lunch today
FTU	false true U eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
FFU	false false U eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
FUU	false unknown U eat rory potato chili pie lunch today	non-factive Paris buy potato-pepper rice for dinner tomorrow
PTU	pfailure true U eat rory potato chili pie lunch today	contrafactive Paris buy potato-pepper rice for dinner tomorrow
PFU	pfailure false U eat rory potato chili pie lunch today	factive Paris buy potato-pepper rice for dinner tomorrow
PUU	pfailure unknown U eat rory potato chili pie lunch today	factive Rory eat potato-chili pie for lunch today

Table 7: One correct input-output pair for each possible combination of input and output tokens given in Table 1. Note that for some input tokens more than one output token is correct. We drop the specialised tokens from the output for brevity.

C Task as Learning an NDFST

The data for the transformer model correspond to a non-deterministic finite state transducer (NDFST). As noted above, the NDFST can be decomposed into a non-deterministic component for producing the correct attitude ascription verb (see the simplified sketch in Fig. 8) and two transducers for producing matching and non-matching embedded clauses respectively.

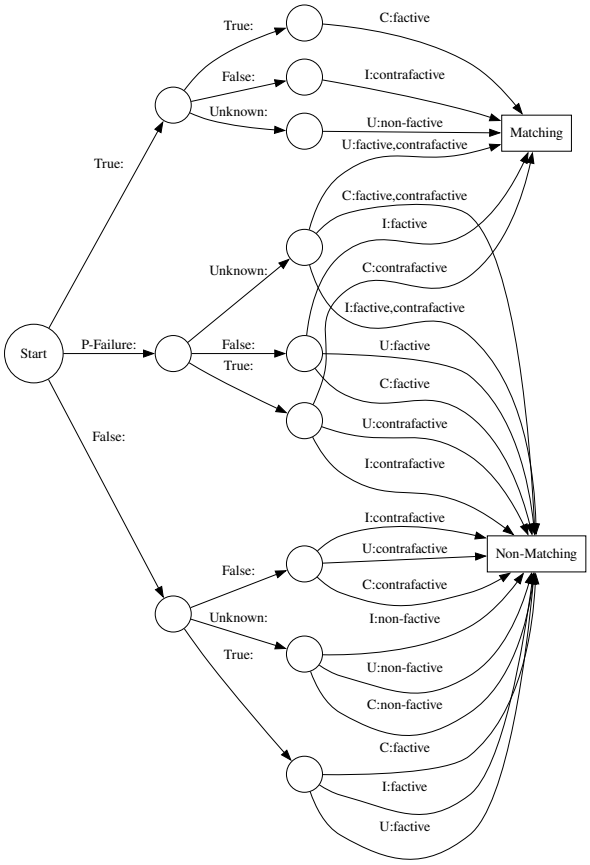


Figure 8: Sketch of a non-deterministic finite state transducer that produces the correct verb for attitude ascription. For simplicity, the permission of more than one output is indicated with a comma. The additional transducers for the embedded clause are indicated by the boxes for matching and non-matching embedded clauses.

The input tokens for the attitude content and the corresponding output tokens for the embedded clauses differ in order (see Fig. 3), but as the transformer models we use have no architecturally encoded order for the input, this order difference does not affect our discussion.

D Linear Programming Details

The max and min are non-linear, making it impossible to directly apply linear programming to our target loss function for creating the various data distributions. To circumvent this problem, we define two helper variables:

1. MaxValue: A variable that is constrained to be greater than or equal to the count of any possible combination, i.e. MaxValue $\geq \#TTC \wedge$ MaxValue $\geq \#FTC \ldots$.
2. MinValue, i.e. A variable that is constrained to be greater or equal to the count of any possible combination, i.e. MinValue $\leq \#TTC \wedge$ MinValue $\leq \#FTC \ldots$.

The target of the linear optimization is then to minimize MaxValue $-$ MinValue.

E Architectural Details

Like Strohmaier and Wimmer (2025), we base our experiments on the standard transformer model included in the PyTorch library (Paszke et al., 2017). As in Vaswani et al. (2017), we use an encoder-decoder architecture. We implemented the models using PyTorch and parallelised using the lightning-Fabric library. For training and evaluation, we used two A100 GPUs.

Like Strohmaier and Wimmer (2025), we use the Binary Cross Entropy (BCE) loss function. Since this is not a typical language modelling task, we have special cases, e.g. with more than one allowed embedded clause, where we set the target value for all allowed embedded clauses to 0.5. Also, if two attitude verbs are allowed, we set the target value for both to 1.

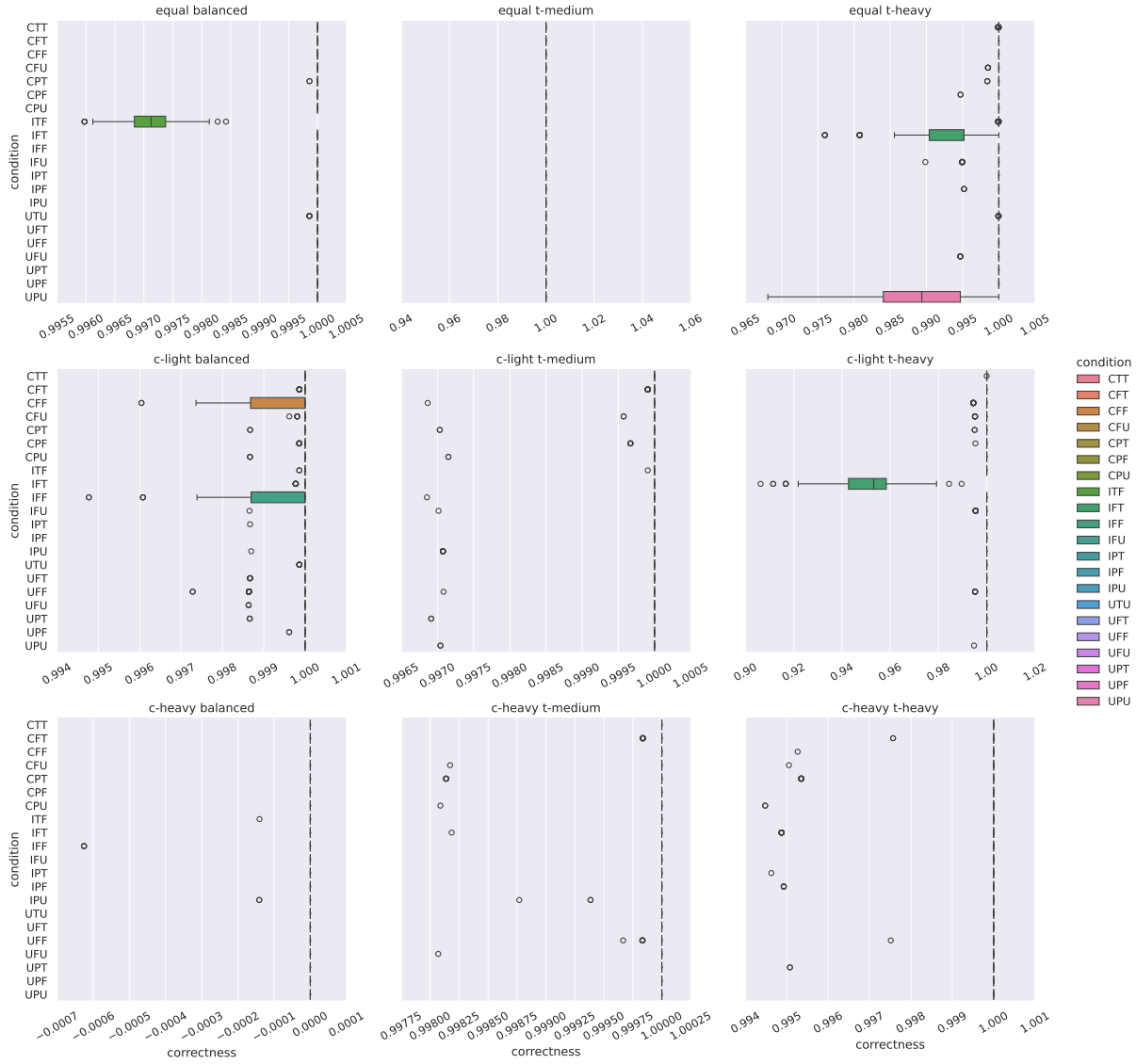
F Hyperparameter Search Details

The target of the hyperparameter search was to maximize the overall correctness of output created on the dev set. All 9×60 settings were fully explored (no pruning). The 60 settings were explored by 4 processes, which communicated via a remote mySQL database.

Name	Lower	Upper	Step Size	Log-Space
Embedding dim.	48	480	24	
Hidden dim.	48	480	24	
# Encoder layers	5	25	5	
# Decoder layer	5	25	5	
Dropout prob.	0.1	0.3	0.1	
Learning rate	1e-07	1e-03	–	✓
Epochs	5	50	1	
Batch size	8	3200	8	

Table 8: Hyperparameter settings used for search by Optuna.

G Learning Trajectory



Performance on the 21 possible conditions given in Table 1.

Significance Test We use the two-sided permutation-test implementation of SciPy (Virtanen et al., 2020) with 9999 resamples and permutation type "samples". The test statistic was the difference between the sum of comparisons in which the model performed better on factives than on contrafactives and the sum of comparisons where the reverse held.

step	f	c	nf	comb
0	9.546%	32.237%	0.027%	11.279%
200	98.280%	83.964%	99.362%	93.685%
420	99.996%	100.000%	100.000%	99.998%
640	100.000%	100.000%	100.000%	100.000%
840	100.000%	100.000%	100.000%	100.000%
1040	100.000%	100.000%	100.000%	100.000%
1260	100.000%	99.991%	100.000%	99.997%
1480	100.000%	100.000%	100.000%	100.000%
1680	100.000%	100.000%	100.000%	100.000%
1880	100.000%	100.000%	100.000%	100.000%
2100	100.000%	100.000%	100.000%	100.000%
2320	99.982%	100.000%	100.000%	99.994%
2520	100.000%	100.000%	100.000%	100.000%
2720	100.000%	100.000%	100.000%	100.000%
2920	100.000%	100.000%	100.000%	100.000%

(a) Trajectory for distribution equal/balanced

step	f	c	nf	comb
0	0.118%	0.005%	51.458%	17.745%
300	96.863%	96.503%	99.995%	97.763%
600	100.000%	100.000%	100.000%	100.000%
900	100.000%	100.000%	100.000%	100.000%
1220	100.000%	100.000%	100.000%	100.000%
1520	100.000%	100.000%	100.000%	100.000%
1820	100.000%	100.000%	100.000%	100.000%
2120	100.000%	100.000%	100.000%	100.000%
2420	100.000%	100.000%	100.000%	100.000%
2720	100.000%	100.000%	100.000%	100.000%
3020	100.000%	100.000%	100.000%	100.000%
3340	100.000%	100.000%	100.000%	100.000%
3640	100.000%	100.000%	100.000%	100.000%
3940	100.000%	100.000%	100.000%	100.000%
4220	100.000%	100.000%	100.000%	100.000%

(b) Trajectory for distribution equal/t-medium

step	f	c	nf	comb
0	6.470%	1.939%	0.000%	2.183%
2220	96.841%	97.175%	99.671%	97.962%
4420	96.786%	99.860%	99.750%	98.812%
6640	98.921%	99.888%	100.000%	99.636%
8860	99.931%	99.986%	100.000%	99.972%
11080	99.986%	100.000%	100.000%	99.995%
13280	99.995%	99.995%	100.000%	99.998%
15500	99.977%	99.986%	100.000%	99.992%
17720	99.977%	99.977%	100.000%	99.992%
19920	100.000%	100.000%	100.000%	100.000%
22140	99.977%	99.921%	99.995%	99.972%
24360	100.000%	100.000%	100.000%	100.000%
26580	100.000%	100.000%	100.000%	100.000%
28780	99.935%	99.935%	99.884%	99.939%
30980	100.000%	100.000%	100.000%	100.000%

(c) Trajectory for distribution equal/t-heavy

step	f	c	nf	comb
0	61.151%	10.916%	0.000%	36.807%
40	61.151%	10.916%	0.000%	36.807%
60	61.151%	10.916%	0.000%	36.807%
100	61.151%	21.688%	0.000%	39.146%
140	61.146%	21.688%	47.956%	49.534%
180	61.146%	21.681%	48.338%	49.618%
200	61.138%	21.695%	48.338%	49.620%
240	61.538%	38.426%	48.763%	53.573%
280	64.731%	43.045%	52.809%	57.279%
300	67.472%	47.514%	56.854%	60.692%
340	86.343%	78.463%	81.806%	83.643%
380	99.569%	99.280%	99.264%	99.448%
420	99.997%	100.000%	100.000%	99.998%
440	100.000%	100.000%	100.000%	100.000%
460	100.000%	100.000%	100.000%	100.000%

(d) Trajectory for distribution c-light/balanced

step	f	c	nf	comb
0	42.654%	5.093%	0.000%	25.158%
20	42.657%	5.093%	0.000%	25.160%
60	40.413%	5.945%	0.783%	24.185%
80	44.969%	14.543%	6.297%	29.850%
100	49.181%	18.629%	13.273%	34.640%
120	53.754%	28.557%	19.080%	40.651%
160	75.072%	56.298%	45.298%	64.562%
180	97.706%	88.000%	74.814%	90.744%
200	99.915%	92.460%	77.980%	93.664%
240	99.481%	91.894%	89.871%	95.934%
260	99.950%	92.585%	100.000%	98.420%
280	99.992%	95.760%	100.000%	99.098%
300	99.992%	99.978%	100.000%	99.995%
340	99.992%	99.978%	100.000%	99.995%

(e) Trajectory for distribution c-light/t-medium

step	f	c	nf	comb
0	4.480%	4.310%	0.089%	2.920%
20	2.050%	7.598%	0.251%	2.256%
60	9.673%	13.353%	7.377%	9.419%
80	18.697%	25.617%	17.778%	19.503%
100	43.512%	43.650%	41.817%	42.950%
140	99.023%	96.623%	99.076%	98.625%
160	99.465%	97.223%	100.000%	99.100%
200	99.502%	99.993%	100.000%	99.706%
220	99.782%	100.000%	100.000%	99.872%
240	99.691%	100.000%	100.000%	99.819%
280	99.920%	100.000%	100.000%	99.953%
300	100.000%	100.000%	100.000%	100.000%
320	100.000%	100.000%	100.000%	100.000%
360	100.000%	100.000%	100.000%	100.000%

(f) Trajectory for distribution c-light/t-heavy

step	f	c	nf	comb
0	55.907%	39.030%	0.000%	33.341%
100	51.340%	63.507%	23.683%	47.625%
200	68.717%	69.152%	48.880%	63.139%
300	95.509%	95.849%	92.343%	94.797%
420	100.000%	100.000%	100.000%	100.000%
520	100.000%	100.000%	100.000%	100.000%
620	100.000%	100.000%	100.000%	100.000%
720	100.000%	100.000%	100.000%	100.000%
820	100.000%	100.000%	100.000%	100.000%
920	100.000%	100.000%	100.000%	100.000%
1020	100.000%	100.000%	100.000%	100.000%
1140	100.000%	100.000%	100.000%	100.000%
1240	100.000%	100.000%	100.000%	100.000%
1340	100.000%	100.000%	100.000%	100.000%
1420	100.000%	100.000%	100.000%	100.000%

(g) Trajectory for distribution c-heavy/balanced

step	f	c	nf	comb
0	32.243%	6.649%	0.000%	16.257%
40	30.873%	31.429%	0.016%	28.068%
60	28.732%	32.060%	0.147%	27.316%
100	41.997%	45.298%	15.857%	40.489%
140	86.567%	87.152%	71.716%	84.168%
180	97.307%	97.474%	86.237%	94.972%
200	97.797%	97.843%	87.003%	95.516%
240	97.902%	96.719%	87.027%	95.344%
280	99.923%	99.944%	87.027%	97.469%
300	99.944%	99.978%	87.027%	97.478%
340	99.916%	99.858%	87.027%	97.441%
380	100.000%	100.000%	87.027%	97.513%
420	100.000%	99.997%	100.000%	99.998%
440	100.000%	100.000%	100.000%	100.000%
460	100.000%	100.000%	100.000%	100.000%

(h) Trajectory for distribution c-heavy/t-medium

step	f	c	nf	comb
0	1.607%	5.404%	0.048%	2.438%
60	27.054%	32.388%	25.079%	28.802%
120	90.888%	94.114%	92.854%	93.088%
200	96.647%	97.351%	97.101%	97.162%
260	97.690%	95.760%	94.115%	96.433%
320	97.869%	99.111%	94.115%	97.878%
380	98.148%	96.499%	94.115%	96.941%
440	98.316%	98.588%	94.115%	97.811%
520	99.114%	96.122%	94.115%	97.095%
580	99.187%	96.653%	94.115%	97.305%
640	99.253%	97.092%	94.173%	97.541%
700	99.865%	98.639%	95.955%	98.705%
780	100.000%	100.000%	100.000%	100.000%
840	99.993%	100.000%	100.000%	99.997%
880	99.993%	100.000%	100.000%	99.997%

(i) Trajectory for distribution c-heavy/t-heavy

Table 9: Learning Trajectory for 9 distributions (factive=f, contrafactive=c, non-factive=nf, comb=combined). "step" refers to the number of training steps taken, which is equivalent to the number of batches seen.

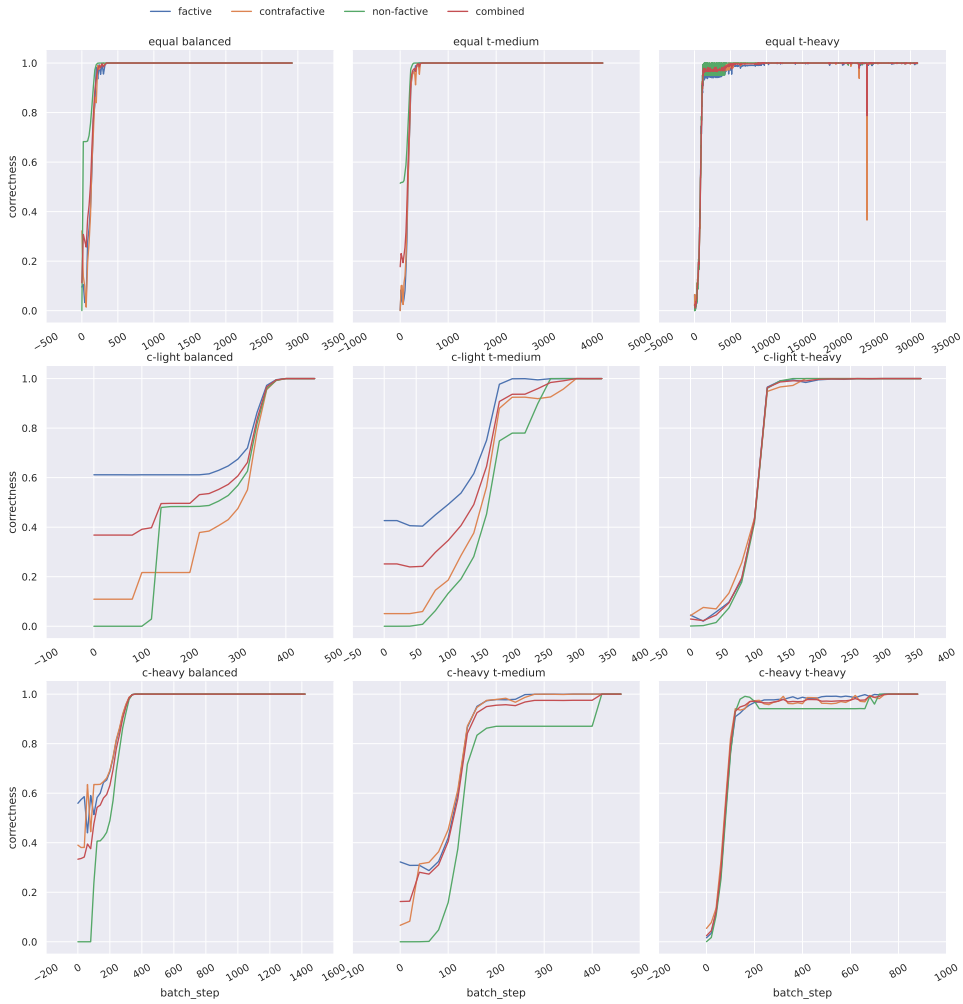
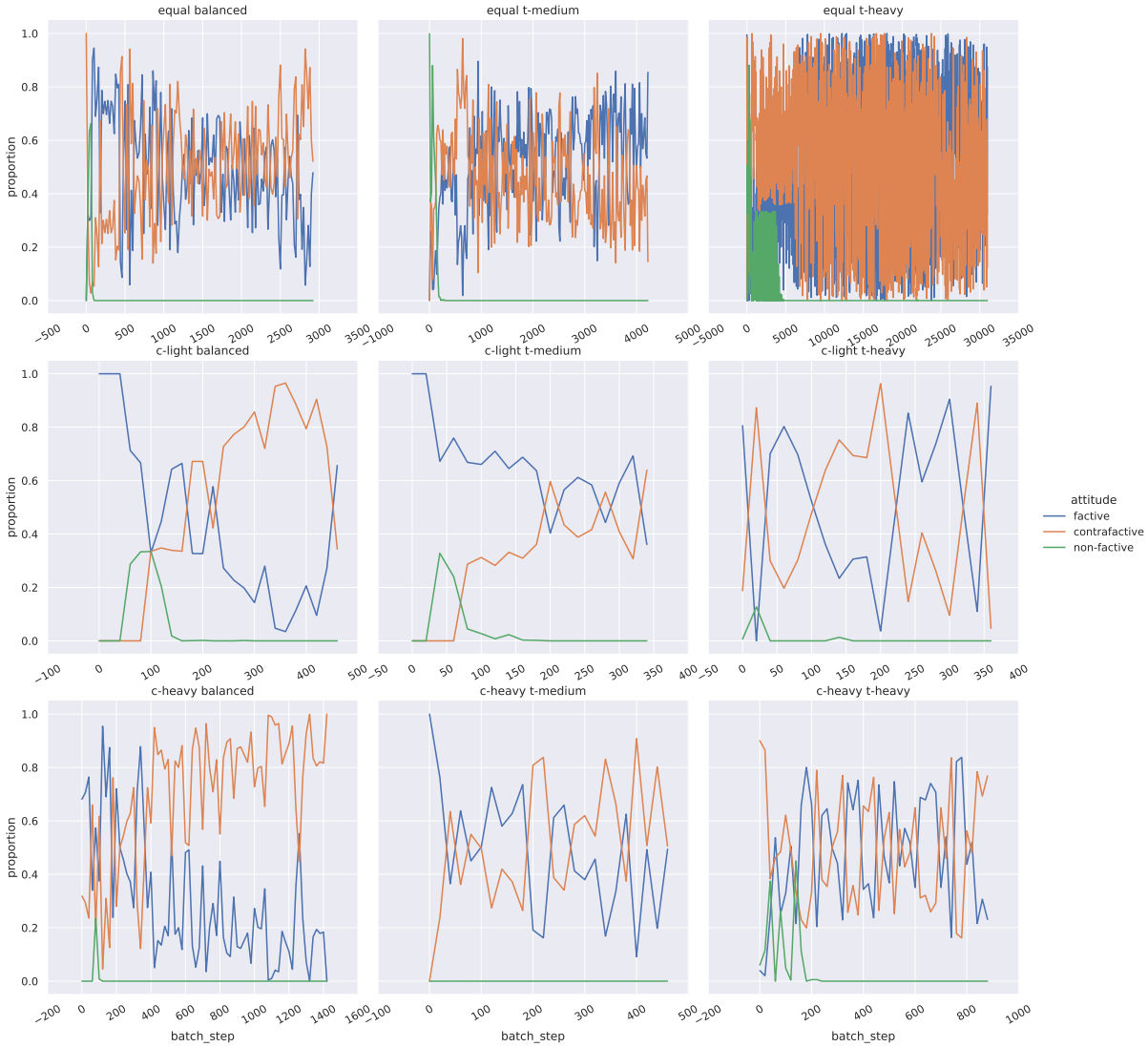


Figure 9: Changes in performance over the training for 3 attitude verbs and 9 data distributions.

H Preference where Factives and Contrafactives Both Permitted



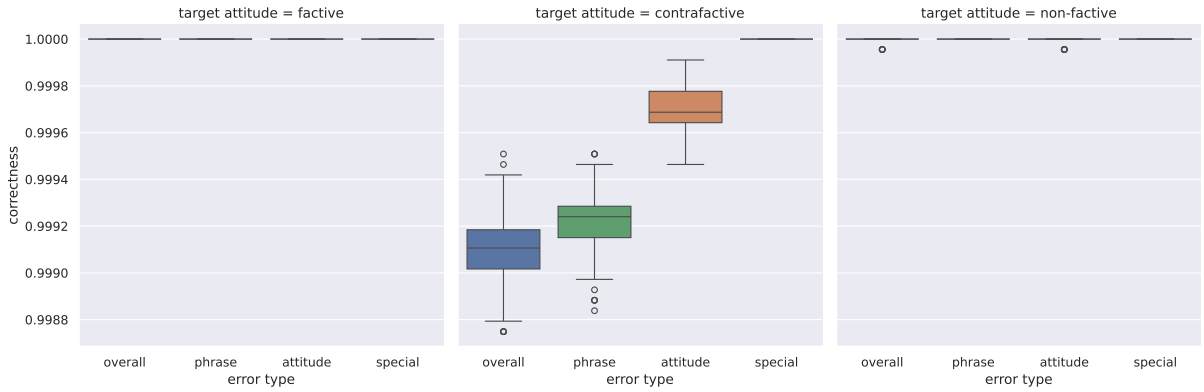
Changes over time in attitude verb preference across target distributions.

I Error Types

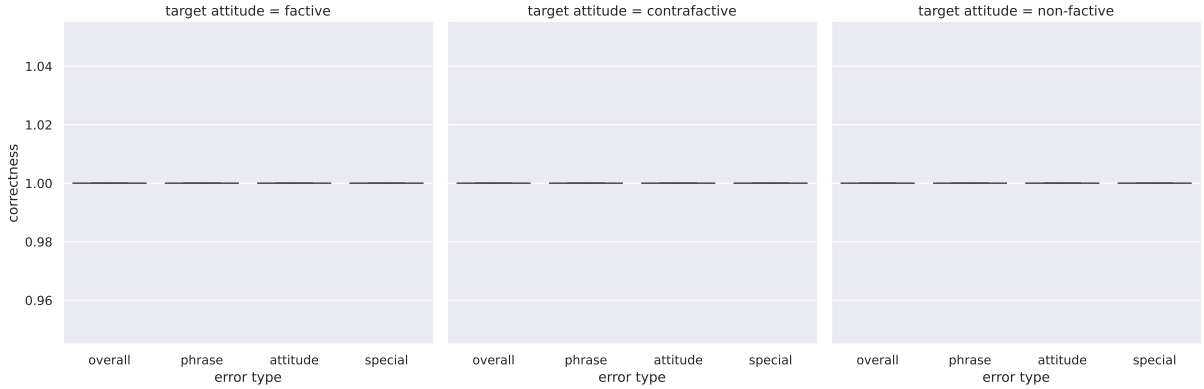
We can distinguish three parts of an output that can be responsible for an error:

1. the tokens of the embedded clause,
2. the token for the attitude verb, or
3. the special tokens

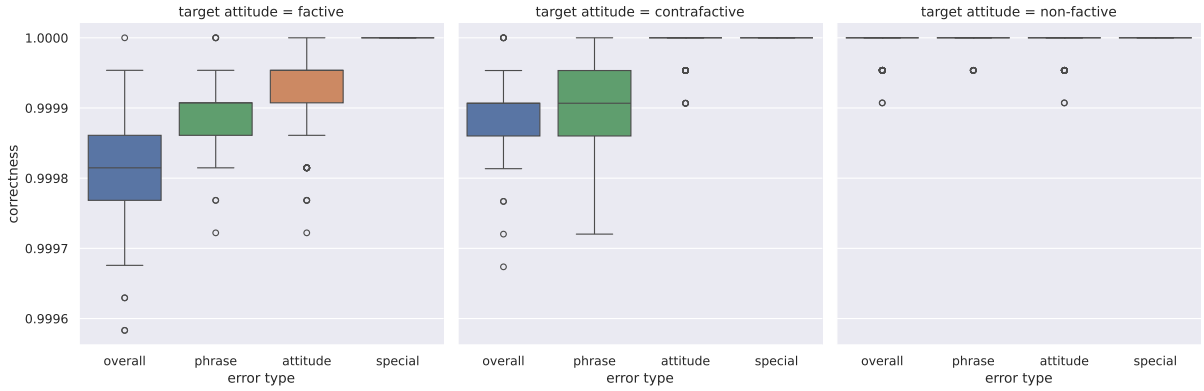
Because the vocabulary is fixed for each token position special token errors are architecturally impossible. That we find no such errors merely reflects the absence of coding errors. Notably, the worst performance on input that allows contrafactives, found in distribution equal/balanced, is primarily due to errors in the production of embedded clause tokens.



(a) Error types for allowed attitude verbs given the distribution equal/balanced.

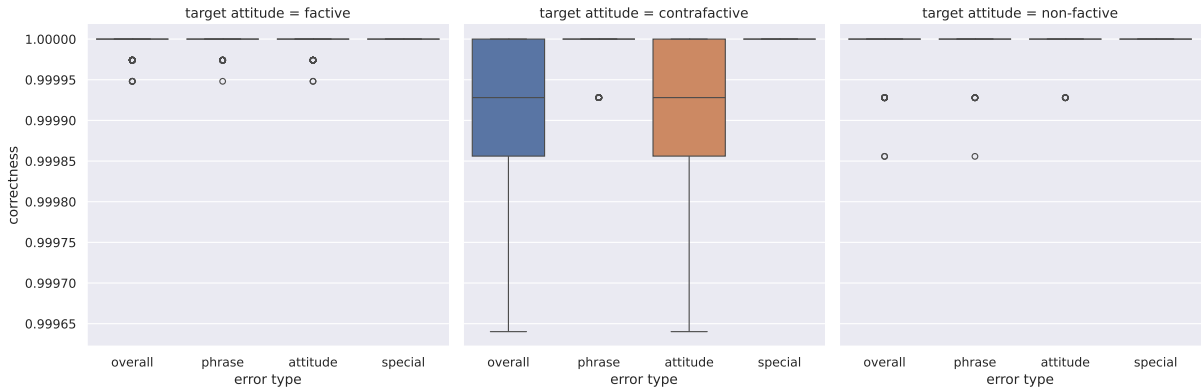


(b) Error types for allowed attitude verbs given the distribution equal/t-medium.

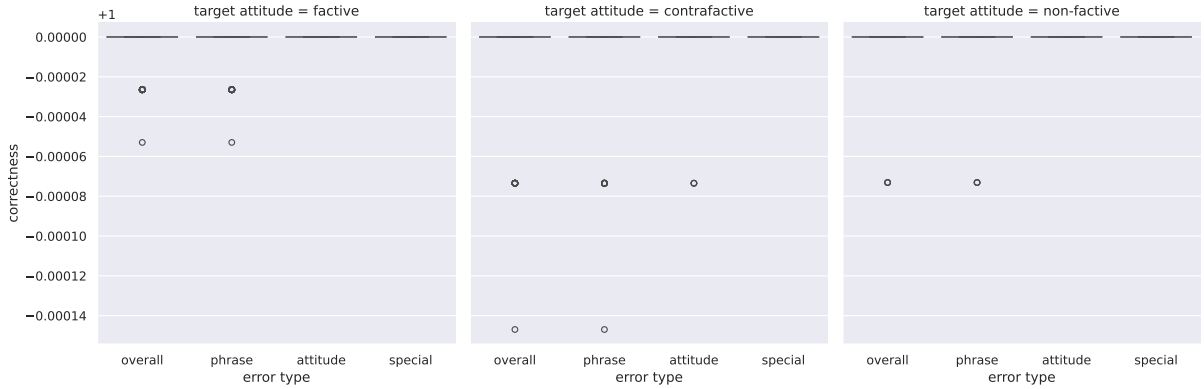


(c) Error types for allowed attitude verbs given the distribution equal/t-heavy.

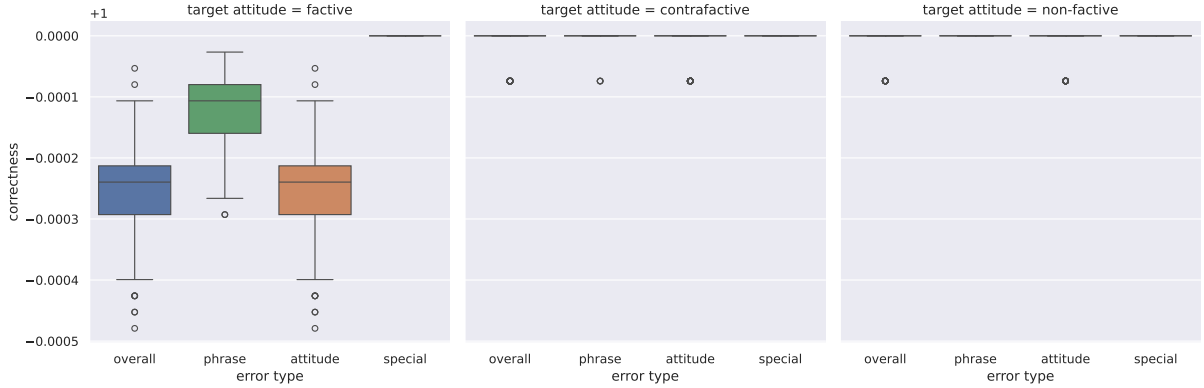
Figure 10: Error types by attitude verb given distributions that are equal.



(a) Error types for allowed attitude verbs given the distribution c-light/balanced.

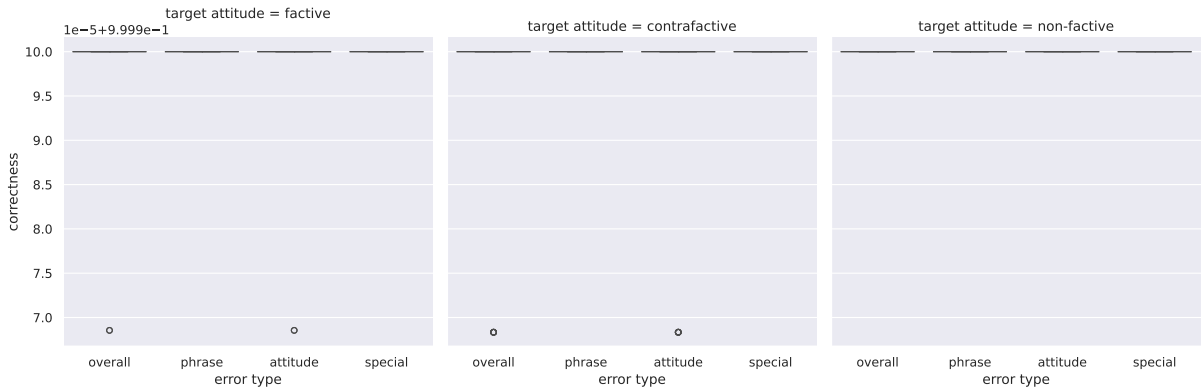


(b) Error types for allowed attitude verbs given the distribution c-light/t-medium.

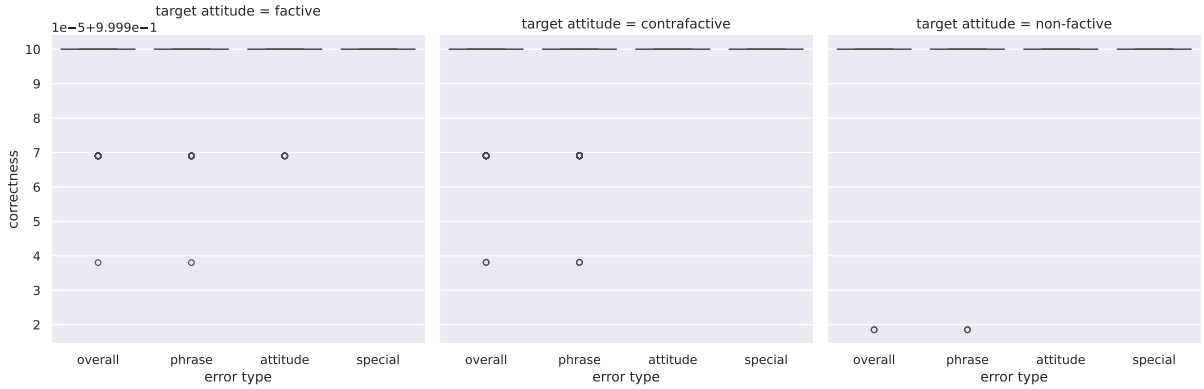


(c) Error types for allowed attitude verbs given the distribution c-light/t-heavy.

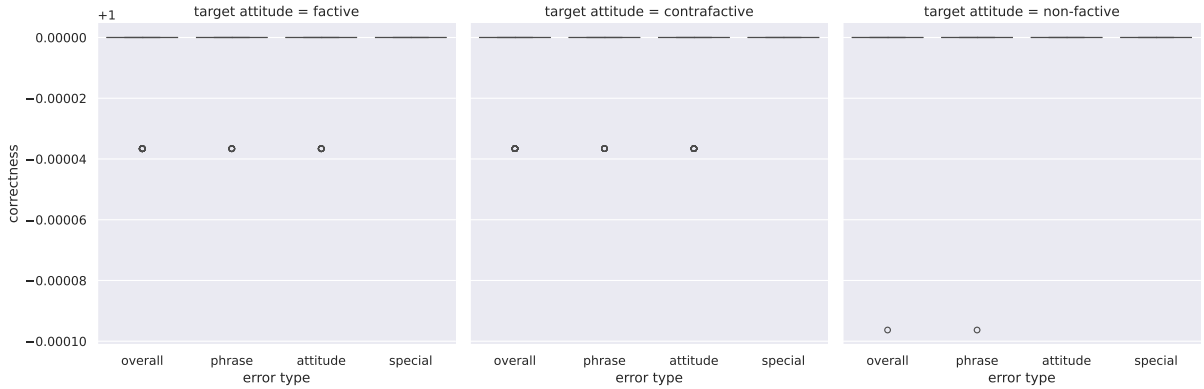
Figure 11: Error types by attitude verb given distributions that are c-light.



(a) Error types for allowed attitude verbs given the distribution c-heavy/balanced.



(b) Error types for allowed attitude verbs given the distribution c-heavy/t-medium.

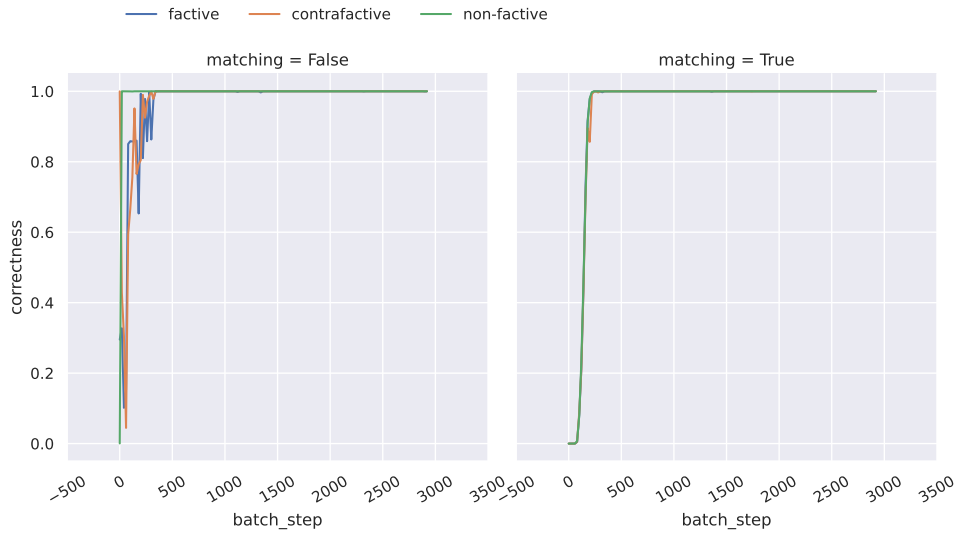


(c) Error types for allowed attitude verbs given the distribution c-heavy/t-heavy.

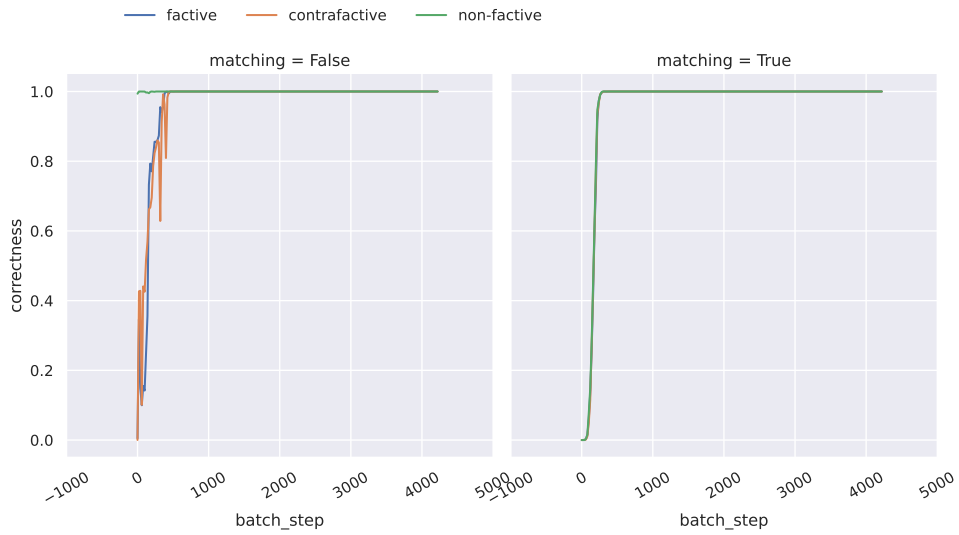
Figure 12: Error types by attitude verb given distributions that are c-heavy.

J Matching vs. Non-Matching

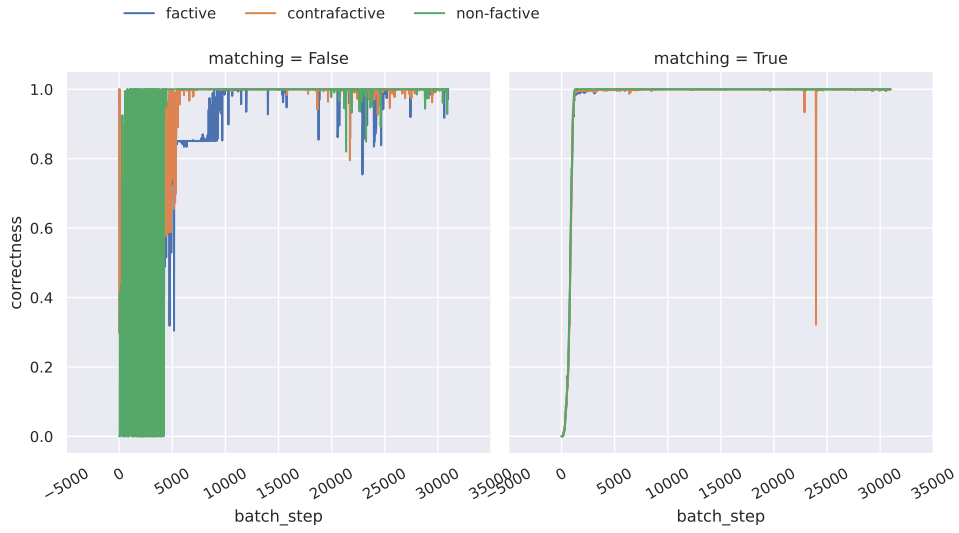
Strohmaier and Wimmer (2025) reported strong differences in performance between conditions that require matching embedded clauses and conditions that require non-matching embedded clauses (see Table 1 to see into which group each of the 21 conditions falls). We provide here the learning graphs split up by what kind of embedded clause is required for all 9 distributions. Our results replicate those of Strohmaier and Wimmer (2025) across distributions.



(a) Matching vs. non-matching performance for allowed attitude verbs given the distribution equal/balanced.

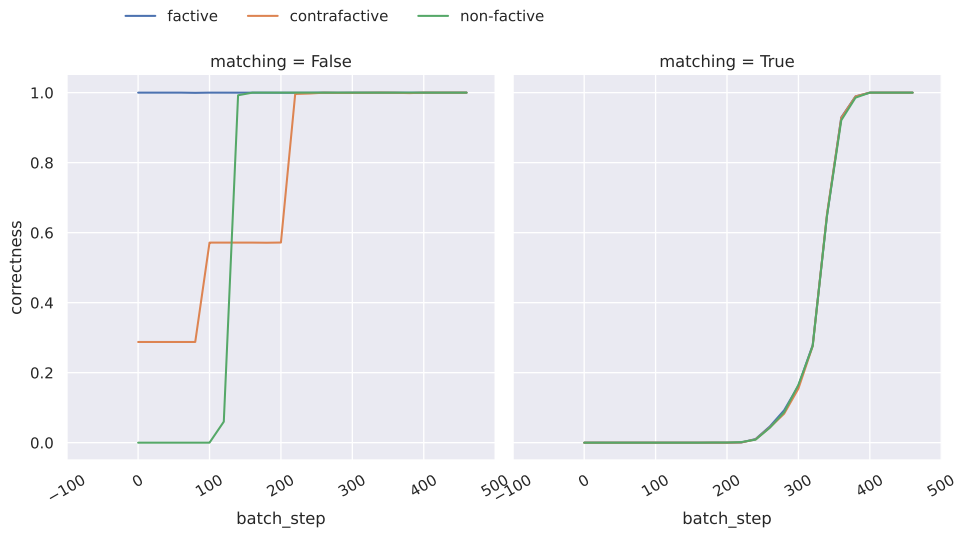


(b) Matching vs. non-matching performance for allowed attitude verbs given the distribution equal/t-medium.

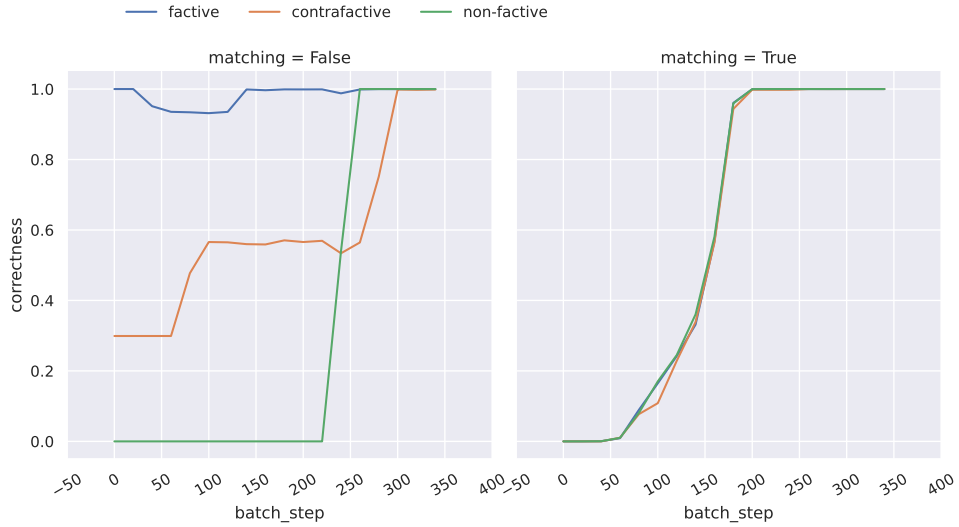


(c) Matching vs. non-matching performance for allowed attitude verbs given the distribution equal/t-heavy.

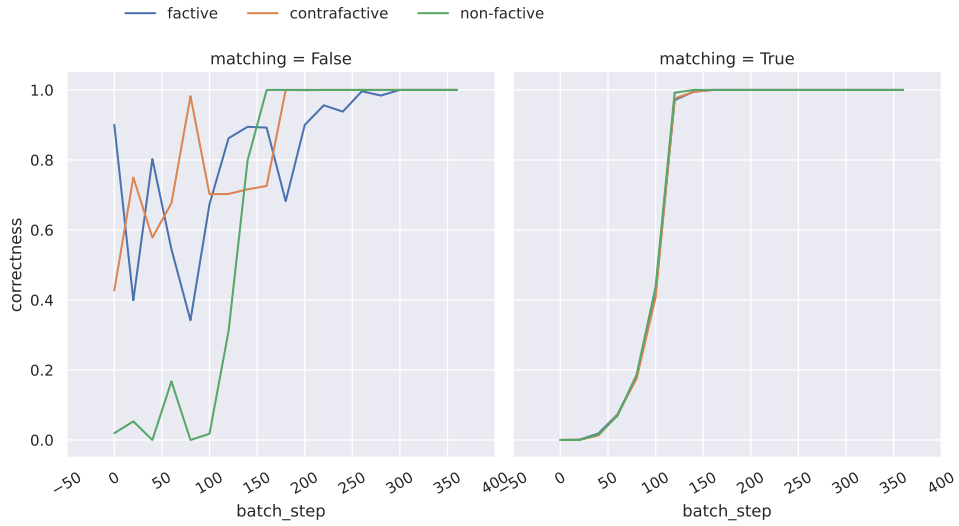
Figure 13: Matching vs. non-matching performance by attitude verb given distributions that are equal.



(a) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-light/balanced.

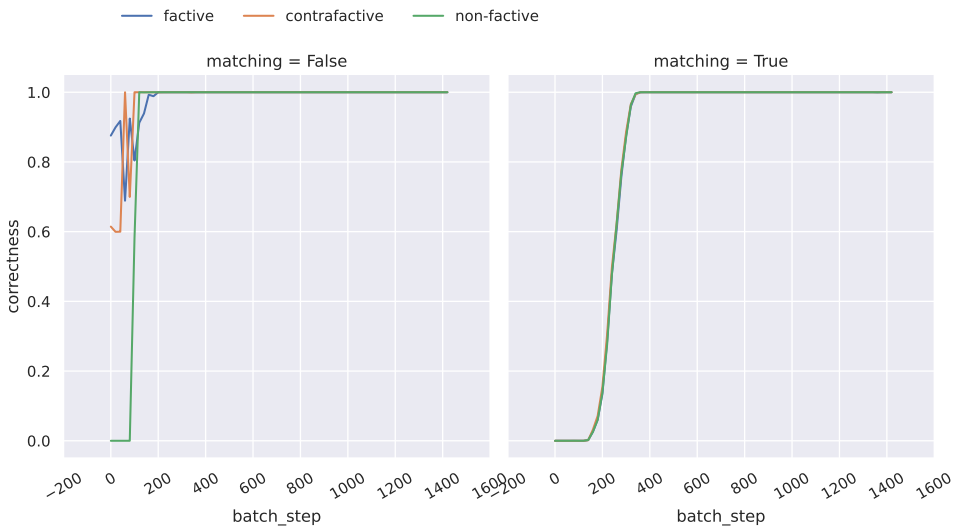


(b) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-light/t-medium.

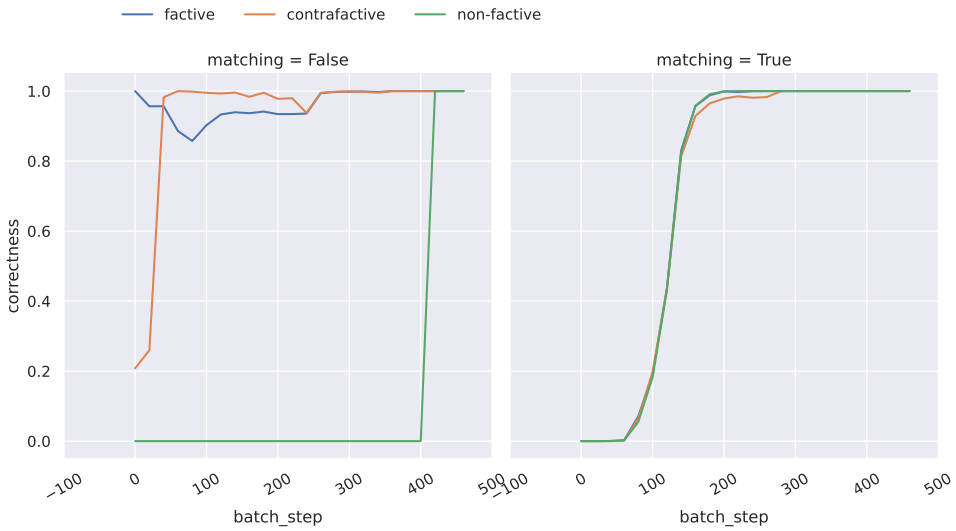


(c) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-light/t-heavy.

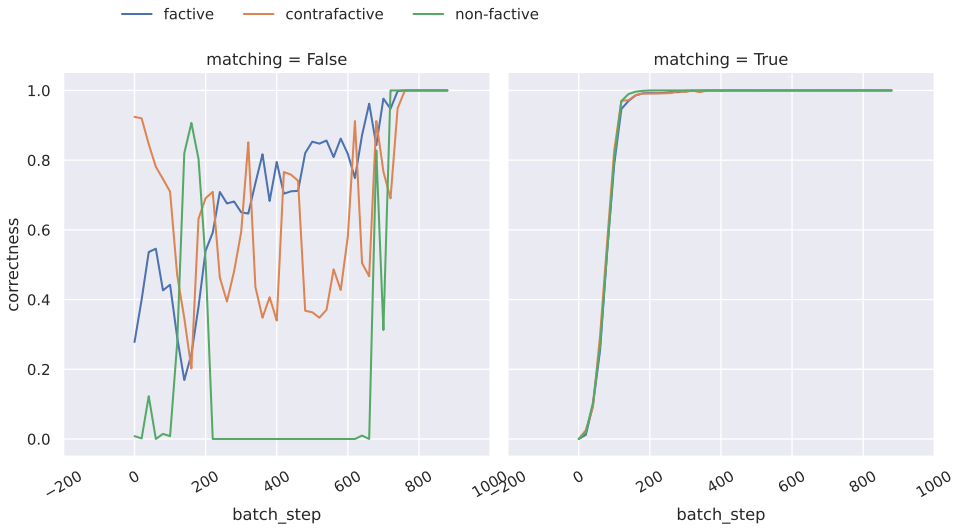
Figure 14: Matching vs. non-matching performance by attitude verb given distributions that are c-light.



(a) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-heavy/balanced.



(b) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-heavy/t-medium.



(c) Matching vs. non-matching performance for allowed attitude verbs given the distribution c-heavy/t-heavy.

Figure 15: Matching vs. non-matching performance by attitude verb given distributions that are c-heavy.