

Processing presupposition: verifying sentences with 'only'

Christina Kim
UCLA
chkim@humnet.ucla.edu

PLC / February 25, 2007

Presuppositions are special

Presuppositions are thought to constitute their own category—distinct from e.g. assertions, conversational implicatures—based on two ways in which presuppositions behave differently from other types of information that can be conveyed by a sentence:

1. Their projection behavior—that is, how the presuppositions of a complex sentence are related to the presuppositions of its subparts.
2. Their status in discourse—presuppositions have a particular information status with respect to a discourse context (noncontroversial, backgrounded, already in the common ground).

My objective here: investigate the latter question from a behavioral perspective.

How do comprehenders treat presuppositional material for the purpose of verifying/falsifying a sentence?

Presuppositions in sentence verification

We'll ask this question experimentally by having people verify/falsify a sentence against a picture—we can then infer what they are doing procedurally from their reaction times (RTs), based on what (visual) evidence we know it would take them to verify/falsify a sentence of a particular type.

- Do comprehenders treat presuppositions and assertions differently at all?
- If so, how are the presuppositions of a sentence verified—procedurally—with respect to the assertion?

(This approach might be seen as a kind of counterpart to the one taken by Chemla & Schlenker (2006), who started with different pairings of presupposition triggers and quantificational environments, and asked whether inferences predicted by various projection theories actually arise.)

Presuppositions in sentence verification

Based on 3 sentence-picture verification experiments involving sentences with 'only', I'll argue for the following picture:

- Comprehenders do treat presupposed and asserted information differently when understanding a sentence.
- Presuppositions are never explicitly verified by comprehenders—they're always assumed to be satisfied.
- If, however, information is encountered in the process of verifying the main assertion of a sentence which is inconsistent with your assumptions about the common ground, this is grounds for rejecting the sentence. Presupposition failure, when it occurs, is 'accidental'.

A point of clarification...

Most theories of presupposition (projection) make no claim whatsoever about how presuppositions should be treated by people in actual conversation/language use.

- Typically, if computation of sentence meaning requires partitioning out separate 'semantic' and 'pragmatic' components (take e.g. Karttunen & Peters' extension and implicature expressions), this is only for the purpose of deriving complex sentence meanings from their constituent parts—that is, at the end of a derivation you have a semantic and pragmatic meaning for a sentence, and nothing more is said about what one does with these components of meaning in order to understand the sentence.
- In dynamic approaches (e.g. Heim context-change potentials; Kamp DRT, Veltman, Zeevat, Beaver update semantics), there does seem to be a claim about procedure—that is, there's a sequence of steps you have to execute in order to finally update your context with the information in a sentence. For Heim, presuppositions are implemented as definedness conditions on context updates: presuppositions are preconditions for truth evaluability, in that an update is possible only when they are satisfied.

A point of clarification...

Most theories of presupposition (projection) make no claim whatsoever about how presuppositions should be treated by people in actual conversation/language use.

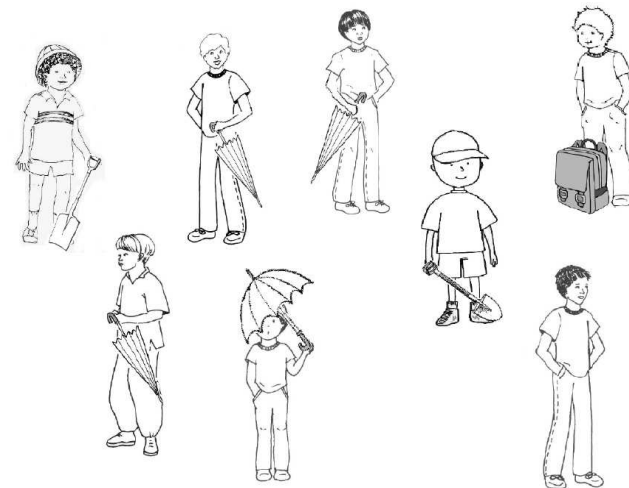
- In fact, such a system still doesn't necessarily predict that language users will adopt this procedure as a comprehension strategy—first verifying that all preconditions are met, and only then proceeding to compute the propositional content of a sentence. One might still think that people tend to simply assume any presuppositions are satisfied, because the alternative is impractical—it wouldn't get you anywhere (cf. von Stechow 2006).

How the experiments work

Different types of sentences (determiners) require different verification procedures.

- (1) *Every kid has an umbrella.*

How the experiments work



How the experiments work

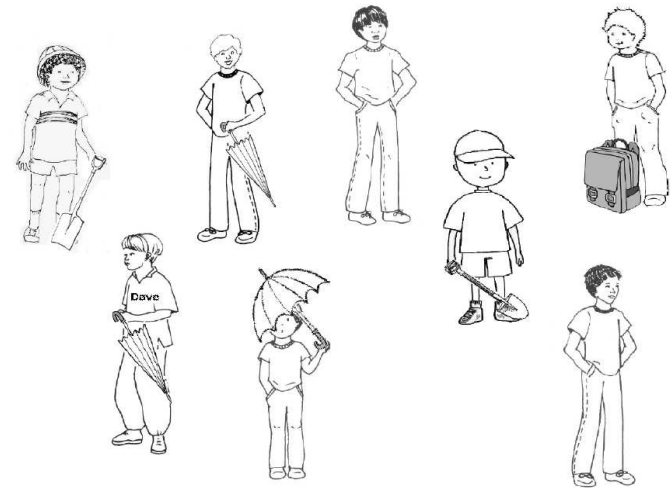
To verify this sentence, you have to go through and check each kid for whether he has an umbrella. (Expect responses to be relatively slow.) In this case, the sentence is true.

- Prediction: True conditions should take longer than False conditions—if the sentence is True, you'll exhaust your search space without finding a falsifier; if it's False, you can stop upon finding a falsifier.

Contrast with:

(2) *Dave has an umbrella.*

How the experiments work



How the experiments work

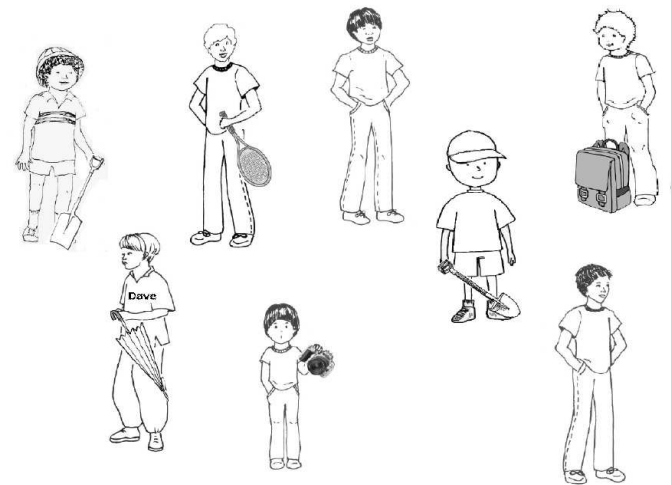
To verify this sentence, just find 'Dave'—responding True or False will take only as long as it takes to find the relevant individual. (Expect responses to be relatively fast.)

- Predict no True/False asymmetry.

Now consider:

(3) *Only Dave has an umbrella.*

How the experiments work



How the experiments work

I'm assuming that (3):
asserts: No one other than Dave has an umbrella
presupposes: Dave has an umbrella

Question: how do comprehenders treat presupposition when verifying a sentence?

How the experiments work

Here are two possible states of affairs:

- *Option 1—Presuppositions as preconditions:* Systematically start by verifying that any presuppositions are satisfied. Then, only if they are, evaluate the assertion.
- *Option 2—Take presuppositions for granted:* Always start by assuming presuppositions are satisfied—that is, start by evaluating the assertion. Then, after the fact, make sure that the presuppositions of the sentence are satisfied, as you've been assuming.

How the experiments work

For sentence (3), option 1 above makes different predictions from option 2.

(3) *Only Dave has an umbrella.*

- *option 1:* Presupposition failure will be noticed right away, since the first thing you do is verify presuppositions. The sentence will be rejected quickly (patterns like '*Dave has an umbrella*').
- *option 2:* Presupposition failure is only noticed after the assertion has been processed. When the presupposition isn't met but the assertion is True, responses will be slow (patterns with True '*Every kid has an umbrella*').

Experiment 1—Truth value judgment

Factors:

- Sentence type (Every/Name/Only)
- Truth value (True/False)
- Presupposition failure (Yes/No)—only 'Only' condition

Details:

- 26 participants; native speakers of North American English (UCLA undergrads)
- 56 trials/run (8 tokens/cell)

For each trial, participants read a sentence displayed on a computer screen. Then they were shown a picture, and asked to respond 'Yes' if the sentence accurately described the picture, and 'No' if it didn't.

Experiment 1 – Truth value judgment

Every

Every boy has a book.

True: 8 boys, each with a book
False: 8 boys, 4 with books, 4 with non-books

Name

Mark has a book.

True: 8 boys: 'Mark' with a book, 7 boys with either books or non-books
False: 8 boys: 'Mark' with a non-book, 7 boys with either books or non-books

Only

Only Mark has a book.

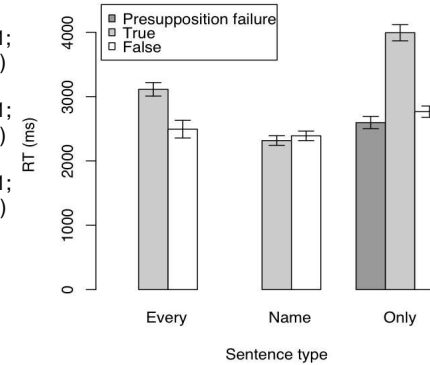
True: 8 boys: 7 boys with non-books, 'Mark' with a book
False: 8 boys: ~4 with books, ~3 with non-books, 'Mark' with a book
Pres. failure: 8 boys: 7 boys with non-books, 'Mark' with a non-book

Experiment 1 – Results

Main effects of

- **Sentence type**
 $F(2, 19) = 44.0, p < .0001$;
 $F(2, 39) = 43.3, p < .0001$
- **Truth value**
 $F(1, 19) = 55.3, p < .0001$;
 $F(2, 39) = 54.5, p < .0001$
- **Presupposition failure**
 $F(1, 19) = 13.7, p < .0001$;
 $F(2, 39) = 13.5, p < .0001$

Sentence type x Truth value interaction ($F(2, 19) = 20.3, p < .0001$; $F(2, 39) = 20.2, p < .0001$)

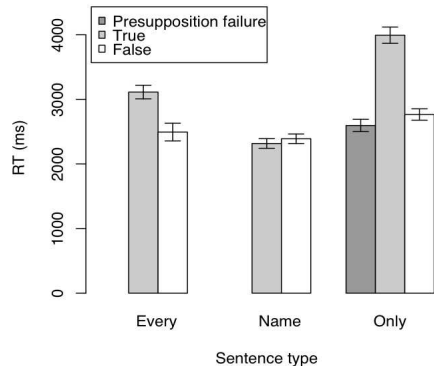


Experiment 1 – Results

1) Every-True > Every-False.

Expected T/F asymmetry since a falsifying item will be found in the False condition; in the True condition you will exhaust the search space without finding a falsifier.

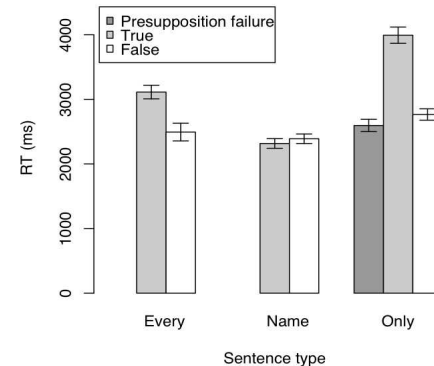
Contrast to the Name conditions (T=F), where no such sequential search is required.



Experiment 1 – Results

2) Only# = OnlyFalse, Only# < OnlyTrue.

Presupposition failure is grounds for rejection just like Falsity is—this looks like support for option 1 above. If preconditions are met, proceed (T or F); if not, abort ('truth value gap'/undefined).



Experiment 1

But—some issues:

- Why is there such a big difference in RTs between Only-True and Every-True?

Suppose people do something like the following:

For Every sentences, domain restriction is trivial (all the individuals are boys, or girls)—apply the same algorithm uniformly (roughly: $\forall x.P(x)$; for each x , does x have property P).

Example: *Every girl has a piece of cake.*

Experiment 1

But—some issues:

- Why is there such a big difference in RTs between Only-True and Every-True?

For Only conditions, first find the 'excluded' individual verify the presupposition. Then subtract that individual from the set to be considered, and apply the same algorithm to the remaining individuals ($\forall x.\neg P(x)$; for each x , does x not have property P).

Example: *Only Lisa has a piece of cake.*

The only difference appears to be the negation in the Only case—seems dubious as the only reason for the big RT difference between Every-True and Only-True.

(We'll return to this question in a bit.)

Experiment 1



Experiment 1

- A more basic problem: what if the mere mention of a name (in e.g. '*Only Lisa has a book*') forces people to attend to/makes salient the corresponding item in the display?

Maybe the Only-Failure cases are being rejected so quickly not because people normally treat presupposed material a certain way, but because drawing your attention to that item happens to also indicate that there is presupposition failure.

Experiment 2 replaces names with definite descriptions ('*the boys*'/'*the girls*'); the intended effect is to make it harder to notice the fact that the 'only' presupposition is violated.

Experiment 2—replace names with definite descriptions

Factors:

- Sentence type (Every/Name/Only)
- Truth value (True/False)

Details:

- 40 native-English participants
- 64 trials/run (8 tokens/cell)

Same procedure as Experiment 1.

Experiment 2

Every

Every kid has a book.

True: 8 boys/girls, each with a book
False: 8 boys/girls, 4 with books, 4 with non-books

Definite

The girls have books.

True: 2 girls with books, 6 boys with either books or non-books
False: 2 girls with non-books, 6 boys with either books or non-books

Only—presupposition not met

Only the girls have books.

True: 2 girls with non-books, 6 boys with non-books
False: 2 girls with non-books, ~3 boys with books, ~3 boys with non-books

Only—strengthened

Only the girls have books.

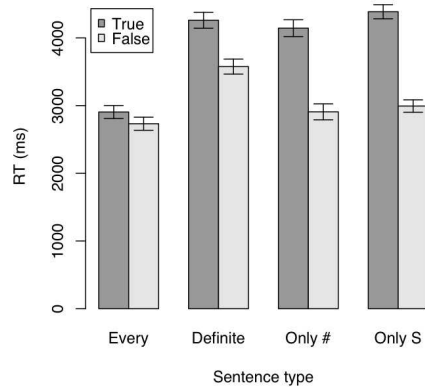
True: 2 girls with books, 6 boys with non-books
False: 2 girls with books, ~3 boys with books, ~3 boys with non-books

Experiment 2—Results

Main effects of

- *Truth value*
 $(F(1,32))=130.5, p<.0001$;
 $F(1,37)=130.9, p<.0001$
- *Sentence type*
 $(F(1,32))=38.7, p<.0001$;
 $F(2,37)=38.8, p<.0001$

Sentence type x Truth value interaction
 $(F(1,3,38))=13.3, p<.0001$;
 $F(2,3,37)=13.2, p<.0001$

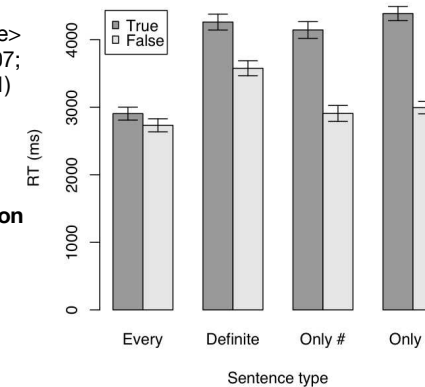


Experiment 2—Results

1) **True > False** for all sentence types (EveryTrue > EveryFalse marginal, $p=.07$; all other adjusted $p<.0001$)

2) **Only-Failure-True (slow) corresponds to the Only-Failure condition (fast) from Expt 1**

So what does that mean?



Experiment 2—Results

- Presuppositions and assertions are treated differently—presupposition failure does not look like falsity (compare Only-Failure-True vs. Only-Nofailure-False).
- Suggests the Expt 1 results are due to task/stimulus-specific things, as suggested above.
- Given such an explanation for Expt 1, the Expt 2 results seem to support option 2 regarding processing presupposition—comprehenders take presupposed information for granted (at least at first).

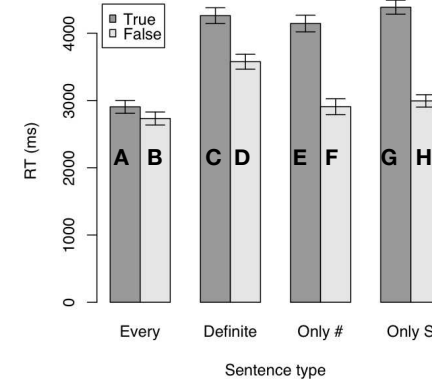
This is why the two Only-True cases are both slow (regardless of the status of the presupposition); cases where the presupposition isn't satisfied are rejected only after the assertion has been processed (almost) in its entirety.

The RTs reflect this in a couple different ways:

Experiment 2—Results

- **E** (Only-Failure-True) ≠ **D** (Definite-False).

Since verifying the presupposition of 'only' will amount to the task required for **D** (e.g. 'The girls have books' is False), you can't be starting out with this step for **E**.



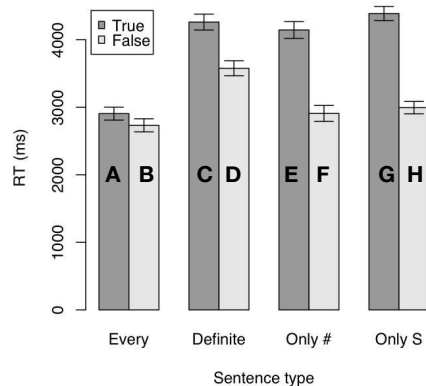
Experiment 2—Results

- **F & H** (the two Only-False conditions) = **B** (Every-False)

Makes sense if these involve the same procedure (universal statement, evaluates to False).

- **H** (Only-Nofailure-False) < **C** (Definite-True)

If you were starting out by verifying the presupposition, we'd expect **C** < **H** (**C** involves one sub-step of **H**).



Experiment 2

- Are people ignoring presuppositions altogether? It might seem so based on the the fact that the two sets of Only conditions look almost the same.

No: people are using the presupposed information because they reject cases where the assertion is True, but the presupposition isn't satisfied.

- A remaining question: It takes a lot longer to respond to 'The boys' that to 'Every kid'.

Why should these be so different? (Both say of some set of individuals that each member has some property.)

Experiment 2

'The boys have books' > 'Every kid has a book'

- *hypothesis 1*: It's got something to do with the definite determiner. After all, definites introduce existence/maximality presuppositions (though these are always satisfied in these stimuli).
- *hypothesis 2*: There is something very different about these cases. For 'Every kid has a book', domain restriction is totally trivial, but for 'The boys have books', it's not—you have to distinguish the set of boys from the set of girls.
 - Only hypothesis 2 makes Definites slow for the same reason that the Only conditions were slow—these are all cases requiring non-trivial domain restriction.

Experiment 3: do sentences like 'Every boy has a book' pattern with 'Every kid has a book' (same quantifier) or with 'The boys have books' (same set restriction requirement)?

Experiment 3—add 'Every boy'/'Every girl'

Sentences:

Every kid has a book.
 Every girl has a book.
 The girls have books.
 Only the girls have books. (presupposition failure)
 Only the girls have books. (no failure)

Details:

- 32 native-English participants
- 80 trials/run (8 tokens/cell)

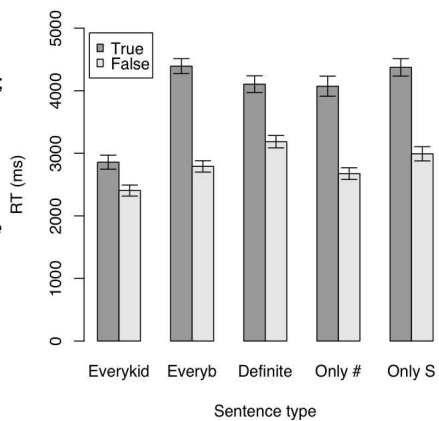
Same procedure as Experiments 1 & 2.

Experiment 3—Results

Main effects of

- *Truth value*
 (F1(1,22))=290.0, p<.0001;
 F2(1,35)=290.4, p<.0001)
- *Sentence type*
 (F1(4,22))=34.5, p<.0001;
 F2(4,35)=34.5, p<.0001)

Sentence type x Truth value
 interaction (F1(4,22))=8.2,
 p<.0001; F2(4,35)=8.6,
 p<.0001)

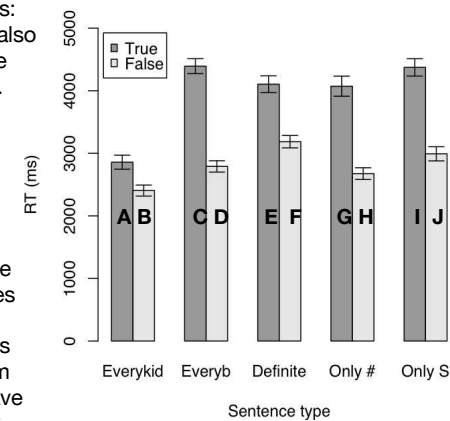


Experiment 3—Results

1) Replicates Expt 2 results: condition **G** is slow (**G>H**, also **G>F**). (T>F for all sentence types, all adjusted p<.005).

2) **C** (Everyboy-True) patterns like **E** (Definite-True), not **A** (Everykid-True).

Supports *hypothesis 2*—the elevated RT for the Definites isn't due to the definite determiner, but because it's hard to apply a non-uniform checking algorithm (you have to evaluate boys differently girls).

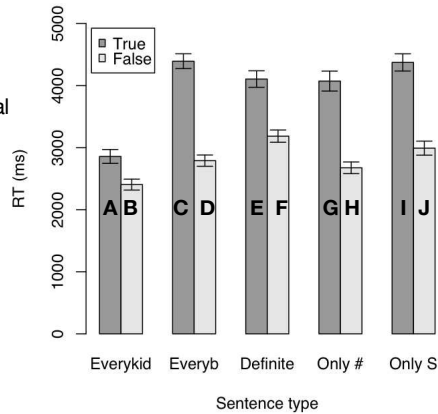


Experiment 3—Results

This gives us an explanation for the big difference between A & B (Every—trivial domain restriction) and all the other conditions.

In the latter cases, you have to perform a more complex procedure:

e.g.
'Only the girls have books'
requires verifying:
'If $\neg \text{girl}(x)$, then $\neg \text{book}(x)$ '



Experiment 3—Results

What we do with presuppositions, according to Expts 2 & 3:

- *Option 2—Take presuppositions for granted:* Always start by assuming presuppositions are satisfied—that is, start by evaluating the assertion. Then, after the fact, make sure that the presuppositions of the sentence are satisfied, as you've been assuming.

Let's try to be a little more precise.

Experiment 3—Results

Here are two variants of option 2 ('take presuppositions for granted'):

1. Start out by assuming for the moment that all presuppositions are satisfied. After verifying/falsifying the assertive component, verify that any presuppositions have been met (as you've been assuming).

→ This would seem to predict that the failure cases will be either the same as or slightly slower than the no-failure cases.

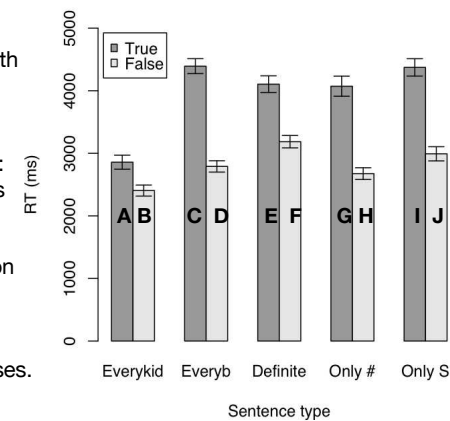
2. Same as above, except there is no separate process that 'verifies presuppositions'. Start out assuming presuppositions are satisfied; go about verifying the assertion. If, somewhere along the way, you happen to discover information that is inconsistent with your assumptions (i.e. some presupposition is violated), stop/reject the sentence. Discovering presupposition violations is 'accidental'.

→ This would predict that the failure condition would be slightly faster (slightly greater probability of being able to terminate early) than the no-failure condition.

Experiment 3—Results

Only-Failure slightly faster than Only-No-failure, for both True ($E < G$, $p < .05$) and False ($F < H$, $p < .05$)

Suggests the latter version: Presuppositions are always assumed to be entailed by the common ground, without separate verification—when a 'presupposition failure' situation arises, it comes about as a byproduct of other processes.



Wrapping up

- Presuppositions and assertions are treated differently in sentence comprehension—they have different behavioral reflexes.
- There is no separate process that verifies that presuppositions are satisfied by the context. Presupposition failure arises only as a byproduct of verifying the assertive content of a sentence, or if the presupposed material is otherwise made extra salient.
- Domain restriction is effortful (and time-consuming). Whether or not restriction is actually/non-trivially required by the context seems to make a difference for the complexity of the verification algorithm.

Wrapping up

- A future direction—this sentence-picture verification paradigm can give us pretty fine-grained information about presupposition-carrying sentences and their presuppositions. If entailments, scalar implicatures, perhaps other inference types turn out to have different behavioral reflexes, we'd have the beginnings of a diagnostic tool.

References

- Chemla, E. & Schlenker, P. 2006. Presuppositions vs. scalar implicatures: an experimental study. Poster presented at OSU.
- Heim, I. 1983. On the projection problem for presuppositions. *WCCFL* 2: 114-126.
- Horn, L. 1969. A presuppositional analysis of 'only' and 'even'. *Papers from the Fifth Regional Meeting of the Chicago Linguistics Society*: 98-107.
- von Stechow, K. 2006. What is presupposition accommodation, again? *Ms.*, MIT.
- Karttunen, L. & Peters, S. 1979. Conventional implicature, in C.-K. Oh & D. Dinneen (eds.), *Syntax and Semantics 11: Presupposition*, Academic Press, NY, 1-56.
- Stalnaker, R. C. 1974. Pragmatic presuppositions. In M. K. Munitz & P. K. Unger (eds.), *Semantics and Philosophy*, NYU Press, 197-213.
- Veltman, F. 1990. Defaults in Update Semantics. In H. Kamp (ed.), *Conditionals, Defaults and Belief Revision*, pp. 28-64. Centre for Cognitive Science, University of Edinburgh.
- Zeevat, H. 1992. Presupposition and accommodation in update semantics. *Journal of Semantics* 9: 379-412.

Appendix

What process(es) is the task actually tapping?

Possibility 1:

You see the sentence and compute the context change potential. Then you get the picture, and do the following: (i) verify that the presuppositions are satisfied, and if so (ii) update the context. Or: (i) start by adding the assertion to the context, and (ii) stop if you detect/discover an inconsistency —e.g. a presupposition isn't satisfied.

Either way, the 2nd reaction time (to the picture) really does indicate something about evaluation of the stimulus sentences given a fixed common ground.

Appendix

What process(es) is the task actually tapping?

Possibility 2:

You see the sentence and update the common ground, accommodating anything necessary (e.g. the existence of a boy named John, or the existence of a set of boys). Then the 1st reaction time (to reading the sentence) would in effect swallow any differences there might be, and there's no way of asking how processes need to be ordered. The 2nd reaction time simply reflects the overall complexity of the updates. All you do when the picture comes up is verify whether the picture shown matches the newly updated context; asserted and presupposed information alike is simply entailed by the common ground.

This predicts differences based on complexity (e.g. the number of properties that need to be satisfied/kept track of), but no difference due to the type/source of the entailment (presupposition failure should have the same effect as falsity).

→ Inconsistent with the data: (i) Only-Failure-True should be similar to Only-NoFailure-False—each only have one thing 'wrong' with them; (ii) Only-Failure-False should be even faster than Only-NoFailure-False—the prior case has both falsity and presupposition failure that make it incompatible with the common ground.

Appendix

As expected, the reaction time to the *sentence* simply reflects the overall complexity of the sentence:

