NELS 44, University of Connecticut
October 18-20, 2013

Incremental Presupposition Evaluation in Disjunction

Aron Hirsch & Martin Hackl

MIT

Contact: *aronh@mit.edu, hackl@mit.edu*

1

The puzzle

- Presupposition projection in **conjunction** shows linear order asymmetries.

2

The puzzle

- Presuppositions project cumulatively out of the first conjunct.

  (1) **The King of Tajikistan** is sick, and Europe is in a financial crisis.

  ➔ *presupposition: there is a King of Tajikistan*

  (2) #**The King of Tajikistan** is sick, and Tajikistan has a king.

  ➔ *presupposition: there is a King of Tajikistan, redundantly asserted, so #*

3

The puzzle

- Presuppositions do not project out of the second conjunct when entailed by the first conjunct.

  (3) Europe is in a financial crisis, and **the King of Tajikistan** is sick.

  ➔ *presupposition: there is a King of Tajikistan*

  (4) Tajikistan has a king, and **the King of Tajikistan** is sick.

  ➔ *no presupposition*

4

The puzzle

- Projection in conjunction (Karttunen 1973):

  > *S1(p) and S2(p') presupposes p & (S1 → p')*

5

The puzzle

- **Disjunction** seems **not** to show linear order asymmetries.

6

The puzzle

- **Disjunction** seems **not** to show linear order asymmetries.

- Classic example, after Partee (2005):

  (5) a. Either there is no bathroom, or **the bathroom** is well-hidden.

     b. Either **the bathroom** is well-hidden, or there is no bathroom.

     → *no presupposition, either order*

- Presuppositions do not project out of either disjunct if entailed by the negation of the other disjunct.

7

The puzzle

- Projection in conjunction:

  > *S1(p) and S2(p') presupposes p & (S1 → p')*

- Projection in disjunction:

  > *S1(p) or S2(p') presupposes (¬S2 → p) & (¬S1 → p')*

- The second clause can affect projection out of the first clause in conjunction, but not disjunction.

- What makes conjunction and disjunction different?

8

2

Why disjunction might be the odd case

- An attractive account for the conjunction asymmetry is to take it as derivative of **general processing considerations**.

  (cf. Schlenker 2008, Schlenker 2009, Chemla & Schlenker 2012)

- This predicts asymmetry also with disjunction.

9

Why disjunction might be the odd case

- Two processing assumptions:

  Incremental context update:
  The common ground is updated with the first conjunct before the second conjunct is processed.

  Incremental presupposition evaluation:
  A presupposition triggered in the first conjunct is evaluated before the second conjunct is processed.

10

Why disjunction might be the odd case

- Presupposition triggered in first conjunct:

  > *[0]* The King of Tajikistan is sick, and *[1]* Tajikistan has a king *[2]*.

- Stages in parse:

  [0]: common ground prior to utterance, C_0

11

Why disjunction might be the odd case

- Presupposition triggered in first conjunct:

  > *[0]* The King of Tajikistan is sick, and *[1]* Tajikistan has a king *[2]*.

- Stages in parse:

  [0]: common ground prior to utterance, C_0

  [1]: the presupposition that Tajikistan has a king, p, is evaluated

  C_0 must entail p (or be adjusted so as to entail p – accommodation).

12

Why disjunction might be the odd case

- Presupposition triggered in first conjunct:

  > *[0]* The King of Tajikistan is sick, and *[1]* Tajikistan has a king *[2]*.

- Stages in parse:

  [0]: common ground prior to utterance, C_0

  [1]: the presupposition that Tajikistan has a king, p, is evaluated

  C_0 must entail p (or be adjusted so as to entail p – accommodation).

  [2]: C_0 updated with S2, $C_0 \cap \{w : \text{Tajikistan has a king}\}$

  Update is redundant.

13

Why disjunction might be the odd case

- Presupposition triggered in second conjunct:

  > *[0]* Tajikistan has a king, and *[1]* the King of Tajikistan is sick *[2]*.

- Stages in parse:

  [0]: common ground prior to utterance, C_0

14

Why disjunction might be the odd case

- Presupposition triggered in second conjunct:

  > *[0]* Tajikistan has a king, and *[1]* the King of Tajikistan is sick *[2]*.

- Stages in parse:

  [0]: common ground prior to utterance, C_0

  [1]: C_0 updated with S1, $C_0 \cap \{w : \text{Tajikistan has a king}\}$

15

Why disjunction might be the odd case

- Presupposition triggered in second conjunct:

  > *[0]* Tajikistan has a king, and *[1]* the King of Tajikistan is sick *[2]*.

- Stages in parse:

  [0]: common ground prior to utterance, C_0

  [1]: C_0 updated with S1, $C_0 \cap \{w : \text{Tajikistan has a king}\}$

  [2]: the presupposition that Tajikistan has a king, p, is evaluated

  p must be entailed by $C_0 \cap \{w : \text{Tajikistan has a king}\}$

  This is the case independent of C_0.

16

Why disjunction might be the odd case

- Processing explanation for conjunction asymmetry …

    When the trigger is in **S1**, p is evaluated before C_0 is updated with S2, so C_0 itself must entail p.

    → *p projects*

    When the trigger is in **S2**, the presupposition is evaluated after C_0 is updated with S1, so if S1 entails p, C_0 does not have to.

    → *p does not project when entailed by the other conjunct*

17

Why disjunction might be the odd case

- Processing assumptions, translated for disjunction:

    Incremental context update:
    The context is (locally) updated with the negation of the first disjunct before the second disjunct is processed (Schlenker 2009).

    Incremental presupposition evaluation:
    A presupposition triggered in the first disjunct is evaluated before the second disjunct is processed.

18

Why disjunction might be the odd case

- Presupposition triggered in first disjunct:

    > *[0]* Either the bathroom is well-hidden, or *[1]* there is no bathroom *[2]*.

- Stages in parse:

    [0]: common ground prior to utterance, C_0

    [1]: the presupposition that there is a bathroom, p, is evaluated

    C_0 must entail p (or be adjusted so as to entail p – accommodation).

    [2]: S2 is now processed, too late to affect projection out of S1.

19

Why disjunction might be the odd case

- Presupposition triggered in second disjunct:

    > *[0]* Either there is no bathroom, or *[1]* the bathroom is well-hidden *[2]*.

- Stages in parse:

    [0]: common ground prior to utterance, C_0

    [1]: C_0 (locally) updated with ¬S1, $C_0 \cap \{w : \text{there is a bathroom}\}$

    [2]: the presupposition that there is a bathroom, p, is evaluated

    p must be entailed by $C_0 \cap \{w : \text{there is a bathroom}\}$

    This is the case independent of C_0.

20

Why disjunction might be the odd case

- **Prediction:**

  When the trigger is in **S1**, p is evaluated without C_0 being updated, so C_0 itself must entail p.

  → *p projects*

  When the trigger is in **S2**, p is evaluated after C_0 is updated with ¬S1, so if ¬S1 entails p, C_0 need not.

  → *p does not project if entailed by ¬S1*

21

Why disjunction might be the odd case

- Prediction from processing assumptions:

  > *S1(p) or S2(p') should presuppose p & (¬S1 → p')*

- Recall, seemingly the empirical fact:

  > *S1(p) or S2(p') presupposes (¬S2 → p) & (¬S1 → p')*

22

Where we are going

- **Three steps:**

  1. Identify a confound in the bathroom example which may make this not a fair test case for asymmetric projection.

  2. Construct unconfounded examples; test those for asymmetric projection.

  3. Return to the confounded examples; show that the processing account predicts asymmetry at an *intermediate* stage of the parse; experimentally test for this asymmetry.

23

Step 1: Confound

> Either there is no bathroom, or **the bathroom** is well-hidden.
> Either **the bathroom** is well-hidden, or there is no bathroom.

- Schema:

  S1 or S2(p), S2(p) or S1

  Non-presupposing disjunct, S1   =   there is no bathroom

  Presupposition, p   =   there is a bathroom

- Not only does ¬S1 entail p, but p also entails ¬S1.

  ¬S1 and p are **equivalent**.

24

6

Step 1: Confound

- Two felicity conditions on a disjunction *S1 or S2(p)* where p projects:

  Condition 1 = presupposition satisfaction

  If p projects, for p to be satisfied, C_0 must entail p.

  Condition 2 = a Gricean constraint on disjunction

  For a speaker to felicitously utter a disjunction, s/he must not be opinionated as to the truth-value of either S1 or S2.

25

Step 1: Confound

- With p and ¬S1 equivalent, Conditions 1+2 conflict (Gazdar 1979):

  $\Rightarrow$ To satisfy p, *S1 or S2(p)* must occur in a C_0 which entails p.

  $\Rightarrow$ Because p entails ¬S1, a C_0 which entails p also entails ¬S1.

  $\Rightarrow$ If C_0 entails ¬S1, the speaker is opinionated that S1 is false.

- Satisfying the presupposition leads to a violation of the Gricean constraint.

- A way out of the paradox: don't project p!

26

Step 2: Avoiding the confound

- Q: Is projection in disjunction asymmetric when the confound is resolved?

27

Step 2: Avoiding the confound

- Q: Is projection in disjunction asymmetric when the confound is resolved?

- Testing ground (following Schlenker 2008):

  S1 or S2(p), *S2(p) or S1*

  ¬S1 entails p, but p does **not** entail ¬S1.

- Effect of **asymmetric** entailment:

  A C_0 which entails p need not entail ¬S1.

  p can project and the Gricean constraint still be respected.

28

7

Step 2: Avoiding the confound

- Examples are difficult to construct and to judge; best attempt:

Suppose that everyone who uses drugs begins at a particular age, different for each drug: anyone who uses cocaine begins using cocaine at age 15; anyone who uses any other drugs (regardless of whether or not they have already used cocaine) begins using those drugs later, after age 20. About John:

*a. Either John has never done cocaine, or he was 15 when he **first** used illicit substances.*

*b. Either John was 15 when he **first** used illicit substances, or he has never done cocaine.*

- Contrast, replicated with >8 informants:

  (a) is easily compatible with a scenario where John did not use any illicit substance. **(b)** has a more robust inference that John used some illicit substance; difficult to judge in a scenario where he used none.

29

Step 2: Avoiding the confound

- Intuition is supportive of asymmetric projection:

  $S1(p)$ or $S2(p')$ presupposes p & $(\neg S1 \rightarrow p')$

- But: these examples are very difficult to judge (please ask!).

- Can we find a more reliable way to test for processing effects?

30

Step 3: A shift in strategy

- In Step 2, we tried to avoid the confound from Step 1.

  Now, we engage with the confound.

- We return to examples where p and ¬S1 are equivalent:

  a. Either there is no bathroom, or **the bathroom** is well-hidden.
  b. Either **the bathroom** is well-hidden, or there is no bathroom.

- And re-evaluate what the processing account predicts for these examples, factoring in the Gricean constraint …

31

Step 3: A shift in strategy

- Trigger in the first disjunct:

  [0] Either the bathroom is well-hidden, or *[1]* there is no bathroom *[2]*.

- Stages in parse:

  [0]: common ground prior to utterance, C_0

32

8

Step 3: A shift in strategy

- Trigger in the first disjunct:

> *[0]* Either the bathroom is well-hidden, or *[1]* there is no bathroom *[2]*.

- Stages in parse:

[0]: common ground prior to utterance, C_0

[1]: the presupposition that there is a bathroom, p, is evaluated.

The interpreter has not yet processed S2 – so:

⇒ there is no way to know that ¬S2 will be equivalent to p.

⇒ there is no way to know that projecting p will violate the Gricean constraint.

⇒ the interpreter assumes that p projects and adjusts C_0 so as to entail it (global accommodation).

33

Step 3: A shift in strategy

- Trigger in the first disjunct:

> *[0]* Either the bathroom is well-hidden, or *[1]* there is no bathroom *[2]*.

- Stages in parse:

[0]: common ground prior to utterance, C_0

[1]: the presupposition that there is a bathroom, p, is evaluated.

⇒ p projects and C_0 is adjusted so as to entail it; global accommodation.

[2]: the interpreter processes S2.

⇒ s/he discovers that ¬S2 is equivalent to p.

⇒ s/he discovers that projecting p has violated the Gricean constraint.

⇒ s/he cancels p; global accommodation is revised to local accommodation.

34

Step 3: A shift in strategy

- **Trigger-first:**

The presupposition projects at an *intermediate* stage of the parse, and is then cancelled at a later stage in the parse.

35

Step 3: A shift in strategy

- Trigger in the second disjunct:

> *[0]* Either there is no bathroom, or *[1]* the bathroom is well-hidden *[2]*.

- Stages in parse:

[0]: common ground prior to utterance, C_0

36

Step 3: A shift in strategy

- Trigger in the second disjunct:

  [0] Either there is no bathroom, or *[1]* the bathroom is well-hidden *[2]*.

- Stages in parse:

  [0]: common ground prior to utterance, C_0

  [1]: C_0 is (locally) updated with ¬S1, $C_0 \cap$ {w : there is a bathroom}

37

Step 3: A shift in strategy

- Trigger in the second disjunct:

  [0] Either there is no bathroom, or *[1]* the bathroom is well-hidden *[2]*.

- Stages in parse:

  [0]: common ground prior to utterance, C_0

  [1]: C_0 is (locally) updated with ¬S1, $C_0 \cap$ {w : there is a bathroom}

  [2]: The presupposition that there is a bathroom, p, is evaluated.

  p does not project for either of two reasons:

  ⇒ p is satisfied independent of C_0 due to update at [1].

  ⇒ S1 has already been processed, so the interpreter knows that p and ¬S1 are equivalent; projecting p will violate the Gricean constraint.

38

Step 3: A shift in strategy

- **Trigger-first:**

  The presupposition projects at an *intermediate* stage of the parse, and is then cancelled at a later stage in the parse.

- **Trigger-second:**

  The presupposition does not project at *any* stage of the parse.

39

Step 3: A shift in strategy

- **Goal:**

  To experimentally test for whether there is an intermediate stage in the parse at which the presupposition projects in trigger-first order, but not trigger-second order.

40

10

Step 3: Experiment

- How willing an interlocutor is to accommodate a presupposition not satisfied in the common ground depends on how **un/expected** the presupposition is:

  (1) I'm sorry I'm late! I had to take **my dog** to the vet.

  (2) I'm sorry I'm late! #I had to take **my llama** to the vet.

41

Step 3: Experiment

- **Overview of experiment:**

  Participants are presented with a bathroom-like disjunction, in context.

  The context is manipulated so as to bias towards an expectation that the presupposition holds, or does not hold.

  If trigger-first order involves global accommodation, and trigger-second order does not, **only** the felicity of **trigger-first order** should degrade in contexts where the presupposition is unexpected.

42

Step 3: Experiment

- **The paradigm (12 items):**

  Condition 1, presupposition likely:
  *John goes to the doctor to be tested for a very **common** disease. The test comes back negative. We conclude: (a) vs. (b).*

  Condition 2, presupposition unlikely:
  *John goes to the doctor to be tested for a very **rare** disease. The test comes back negative. We conclude: (a) vs. (b).*

  Target disjunction, both orders:
  *a. Either John never had the disease, or he **recovered** from it.*
  *b. Either John **recovered** from the disease, or he never had it.*

43

Step 3: Experiment

- **Predicted judgments:**

  Condition 1, presupposition likely:
  *John goes to the doctor to be tested for a very **common** disease. The test comes back negative. We conclude: ✔(a), ✔(b).*

  Condition 2, presupposition unlikely:
  *John goes to the doctor to be tested for a very **rare** disease. The test comes back negative. We conclude: ✔(a), ?#(b).*

  Target disjunction, both orders:
  *a. Either John never had the disease, or he **recovered** from it.*
  *b. Either John **recovered** from the disease, or he never had it.*

44

Step 3: Experiment

- **Task = forced choice.**

  Participants are presented with either the likely context or the unlikely context from a given item in a Latin Square design; the context is followed by the target disjunction in both orders.

  Participants select which order sounds to them most natural in context.

- **Prediction in task:**

  There should be a lower relative proportion of trigger-first responses in the unlikely condition than in the likely condition.

- 64 participants, run on Amazon Mechanical Turk, $0.20/HIT.
  (experimental scripts: Gibson et al. 2011, Erlewine & Kotek 2013)

45

Step 3: Experiment

- Results show a significant effect of condition on preferred order ($|z| = 2.68$)

- Plot, proportion of trigger-second vs. trigger-first responses by condition:
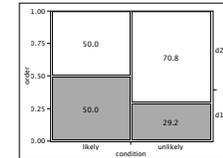
- **Likely condition (baseline):**
  Trigger-first order and trigger-second order equally preferred.

- **Unlikely condition:**
  Trigger-first order is relatively dispreferred to trigger-second order.

46

Conclusion

- Left-to-right asymmetries in the way presuppositions project in complex sentences are general effects of **incremental processing**.

- Three steps in understanding disjunction:

  1. Disjunctions which are globally symmetric are confounded.

  2. Unconfounded disjunctions appear globally asymmetric.

  3. Experimental results from confounded disjunctions show that there is projective asymmetry at an intermediate stage in parsing.

- Confounded disjunctions which at first sight look like a problem case for a processing approach become its strongest evidence.

47

Acknowledgements

48

12