Utrecht University

Generating pragmatically appropriate sentences from propositional logic:
the case of conditional and biconditional

Renhao Pei

Based on master thesis supervised by Kees van Deemter

@LACompLing2021
December 15, 2021

Generating natural language from logical formulas

To name a few applications:

- Explaining the output of a reasoning engine (Coppock and Baxter, 2009)
- Explaining the output of a logistic planning system (Kutlak and van Deemter, 2015)

- Providing feedback to students of logic (Flickinger, 2016)

Generating natural language from logical formulas

As Mayn and van Deemter (2020) put it:

…the meaning of logical connectives is not always the same as that of their natural language counterparts…

$\rightarrow$: If…then…,

$\leftrightarrow$: If and only if…, then…

But sometimes, mismatches exist.

Conditional Perfection(CP)

CP is a well-known mismatch that has been intensively discussed since Geis and Zwicky (1971).

P	Q	$P \rightarrow Q$	$\sim P \rightarrow \sim Q$	$(P \rightarrow Q) \wedge (\sim P \rightarrow \sim Q)$	$P \leftrightarrow Q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	T	F	F	F
F	F	T	T	T	T

- **Sometimes** when people say *if…then…,* they mean *if and only if…then…:*
- A. If you mow the lawn, (then) I'll give you 5 euro.

  A'. If you don't mow the lawn, (then) I won't give you 5 euro.

  B. If and only if you mow the lawn, (then) I'll give you 5 euro.

- *A* invites an inference of *A'*, thus conveying the conjunction of *A* and *A'*, namely *B*.

Conditional Perfection(CP): does it always happen?

- C. If you mow the lawn, (then)you will calm down.

  ?C’. If you don’t mow the lawn, (then) you won’t calm down.
- *C* won’t invite inference like *C’*.

- Inducements(promises&threats) vs. Advice(tips&warnings)
- *C* is advice whereas *A* is an inducement.

- A major difference between inducements and advice is whether the speaker has **control over the consequent**.

Conditional Perfection(CP): when does it happen?

- CP is more likely to happen in **inducements**, and less likely to happen in **advice**, as shown in several experiments (Evans & Twyman-Musgrove, 1998; Newstead, 1997; Ohm & Thompson, 2004).

- When CP happens, *if…then…* would mean $\leftrightarrow$ (Mismatch).

- Then how do we deal with such mismatch?

Cancelling CP

- When CP is present, it can still be cancelled (Herburger, 2016):

…Conditional Perfection is not tied directly to the semantics of conditionals but is rather a pragmatic phenomenon.

- A. If you mow the lawn, I'll give you 5 euro.
- D. If you mow the lawn, I'll give you 5 euro. If you don't want to mow the lawn, you can water the flowers, I'll give 5 euro as well.

- p $\rightarrow$ q: If p then q, <u>if not p then might still q.</u> (Cancelling CP)

Taking advantage of CP

- We may also want to take advantage of CP:
- Since *if…then…* would mean $\leftrightarrow$, if CP is present,

Instead of…

- p $\leftrightarrow$ q: If and only if p, then q.

We can simply say…

- p $\leftrightarrow$ q: If p, then q.

(shorter and sounds more natural)

Conditional Perfection(CP): summary

- For *advice:* no CP

Summary:

Label	FOL	Natural Language
Inducements (CP)	$P \rightarrow Q$	If P then Q, **if not P then might still Q.** (Cancelling CP)
	$P \leftrightarrow Q$	**If** P, then Q. (Utilizing CP)
Advice (No CP)	$P \rightarrow Q$	If P, then Q.
	$P \leftrightarrow Q$	If and only if P, then Q.

Experiment: the pipeline

- 0. An atomic proposition bank & a knowledge base for consequent
- 1. **Generation** of binary propositions out of an atomic proposition bank
- 2. **Classification** based on properties of the consequent

Step 1 and 2 create the input for the algorithm (FOL & label)

- 3. **Realization** of the formula into English sentences, according to the classification labels

Experiment: game setting & generation

- Setting: A multiplayer strategy game, in which players can attack, trade with or form alliance with other players.
- The **proposition bank** contains atomic propositions that describe actions in the game involving two players (the speaker and the hearer). It is divided into an antecedent sub-bank and a consequent sub-bank.
- The **knowledge base** contains information about: (a) whether the consequent is desirable and (b) whether the speaker has control over the consequent.
- Antecedents are designed to be neutral (creating minimal pairs for comparison).
- binary logical formula are randomly generated selecting an antecedent, a connective and a consequent.

Experiment: classification

- The labelling criteria

p	whether the speaker has control over q	whether q is desirable for the hearer	Label for p connetive q	
inherently neutral	+ control	+ desirable	promise	inducement
	+ control	- desirable	threat	
	- control	+ desirable	tip	advice
	- control	- desirable	warning	

Experiment: input for the realization

- a. You destroy the bridge $\rightarrow$ I will attack you : ‘inducement(threat)’
- b. You destroy the bridge $\rightarrow$ Player C will attack you : ‘advice(warning)’
- c. You destroy the bridge $\leftrightarrow$ I will attack you : ‘inducement(threat)’
- d. You destroy the bridge $\leftrightarrow$ Player C will attack you : ‘advice(warning)’

Experiment: the (pragmatic) algorithm

```
for l in L:
        if l is in the form of p → q:
                if l has the label ‘inducement’:
                        r = ‘If p, q, but if not p, might still q.’
                elif l has the label ‘advice’:
                        r = ‘If p, q’
        elif l is in the form of p ↔ q:
                if l has the label ‘inducement’:
                        r = ‘If p, q.’
                elif l has the label ‘advice’:
                        r = ‘If and only if p, q’
        R.append(r)
return R
```

Experiment: the baseline algorithm

```
for l in L:
        if l is in the form of p → q:
                r = 'If p, q'
        elif l is in the form of p ↔ q:
                r = 'If and only if p, q'
        R.append(r)
return R
```

Experiment: output comparison

- Baseline:

a. If you destroy the bridge, I will attack you.

b. If you destroy the bridge, player C will attack you.

c. **If and only if** you destroy the bridge, I will attack you.

d. If and only if you destroy the bridge, player C will attack you.

- Pragmatic:

a. If you destroy the bridge, I will attack you, **but if you don't, I might still do**.

b. If you destroy the bridge, player C will attack you.

c. **If** you destroy the bridge, I will attack you.

d. If and only if you destroy the bridge, player C will attack you.

Evaluation: metrics

- Evaluation metrics: faithfulness and naturalness

For faithfulness:

- **Truth table task** in which participants are given a message and asked to indicate which cases are **consistent** with that message(adapted from Sevenants (2008))

For naturalness: a linear scale for 1 (very unnatural) to 5 (very natural)

Evaluation: questionnaire

4a. Sophie sent a message to Hans: 'If you destroy the bridge, I will attack you, and if you don't, I might still do' *

Having recieved Sophie's message,

Tick all that apply.

- [ ] Hans destroyed the bridge and Sophie attacked Hans.
- [ ] Hans destroyed the bridge and Sophie didn't attack Hans.
- [ ] Hans didn't destroy the bridge and Sophie attacked Hans.
- [ ] Hans didn't destroy the bridge and Sophie didn't attack Hans.

4b *

Does the wording of Sophie's message sound natural to you?

Mark only one oval.

	1	2	3	4	5	
very unnatural	◯	◯	◯	◯	◯	very natural

Evaluation: questionnaire & participant

- The questionnaire contains 2 (baseline and pragmatic) * 2 (promise and threat) * 2 (conditional and biconditional) = 8 target messages, + 8 filler messages, hence 16 messages in total.

- Participants: 10 proficient English speakers, 20-40 years old, who don’t know about propositional logic.

Evaluation: results

p	q	p→q	p↔q
T	T	T	T
T	F	F	F
F	T	T	F
F	F	T	T

$p \rightarrow q$

	TT	TF	FT	FF	average accuracy	naturalness
baseline algorithm	1.0	0.95	0.2	0.85	0.75	4.3/5
pragmatic algorithm	1.0	0.9	0.9	0.8	0.9	3.65/5

$p \leftrightarrow q$

	TT	TF	FT	FF	average accuracy	naturalness
baseline algorithm	0.95	1.0	0.95	0.85	0.9125	3.2/5
pragmatic algorithm	0.1	0.95	0.8	0.9	0.9125	4.45/5

Evaluation: results

- Overall scores:

	naturalness score	overall accuracy
baseline algorithm	3.75/5	0.83125
pragmatic algorithm	4.05/5	0.90625

- The designed pragmatic algorithm is better in terms of both faithfulness and naturalness!

Next steps

- To investigate other connectives such as $\vee$ and $\wedge$, thus covering all the connectives used in propositional logic.
  - Exclusive/inclusive or
  - Interaction and transformation between connectives

- Notably, $\neg p \rightarrow q$ and $p \vee q$ are considered as logically equivalent, but they might not be equivalent if realized in natural language, when speech act is taken into consideration:
- Van & Franke (2012) pointed out that: $\neg p \rightarrow q$ can make both promises and threats, but $p \vee q$ can only make threats, not promises.

Thank you!

Utrecht University

Sharing science,
shaping tomorrow