

Bridge inference calculations

Representation

- The bridge semantic representation consists of description logic-based roles and concepts, extended with contexts. Each piece of text is represented by a set of assertions derived from the syntactic LFG dependency representation.

Representation of content and contexts

- After the LFG dependency parsing (f-structure), the resulting structure gets converted to a [clausal format] and further normalize. The assertions are subdivided in three parts: content, context and temporal structure.
- The content clauses mainly contain subconcept and role assertions. Subconcept assertions state to which ontological types (hypernyms) the terms (skolems) belong. Role assertions describe the relation between the various subconcepts. They correspond to the dependencies in the f-structure.

Use of contexts

- The use of contexts enables flat representations for embedded predicates
- They are also used as scope markers, modal sentential adverbs
- Partial representation of Bill said that Ed wanted to talk

```
context(t)
context(ctx(talk:29))
context(ctx(want:19))
top_context(t)
context_relation(t,ctx(want:19),crel(Topic,say:6))
context_relation(ctx(want:19),ctx(talk:29),crel(Theme,want:19))
```

Temporal Structure

- The temporal structure describes the temporal relations between the terms and between the terms and the utterance time.

Subsumption/specificity relations

They are calculated for quantifiers, (intersective) adjectives, hypernymic relations of nominals (exploiting WordNet) and temporal adjuncts.

Implemented in prolog

Specificity computation for nouns, names and verbs, example

Alignment

Text:

Kim hopped.

Hypothesis:

Someone moved.

Specificity
computation

Text:

Kim hopped.

Hypothesis:

Someone moved.

Red: more specific; blue: less specific

Specificity computation for adjectives and quantifiers, example

Specificity computation

Adjectives

Every (↓) (↑) Some (↑) (↑)

Text: Every boy saw a small cat.
Hypothesis: Every small boy saw a cat.

Text: Every boy saw a small cat.
Hypothesis: Every small boy saw a cat.

Text: Every boy saw a small cat.
Hypothesis: Every small boy saw a cat.

Temporal calculations

- Temporal modification affects monotonicity-based inferences and interacts with quantification and negation. But the effect of modifier addition and dropping is different from that found with other modifiers.

Every boy bought a toy from Ed.
entails: Every boy bought a toy.

Last year, in July, he visited us every day.
does not entail: Last year he visited us every day.

Every boy bought a toy.
does not entail:
Every boy bought a toy from Ed.

Last year he visited us every day.
entails
Last year he visited us every day in July.

Contexts and relations between contexts

Negation and various types of modalities are represented as contexts. The relations between the contexts specify the type of modality. Up to now the system has been used to model negation and veridicality relations.

Examples of the veridicality mark-up

Factives

Positive **++/+** forget that forget that X ==> X, not forget that X ==> X

Negative **+/-** pretend that pretend that X ==> not X, not pretend that X ==> not X

John forgot that he had put his keys on the table.

John didn't forget that he had put his keys on the table.

Mary pretended that she had put her keys on the table.

Mary didn't pretend that she had put her keys on the table.

Implicatives

Two-way implicatives *++/- manage to manage to X ==> X, not manage to X ==> not X*
+-- fail to fail to X ==> not X, not fail to X ==> X

++ force to force X to Y ==> Y

One-way implicatives *+ prevent from prevent X from Y ==> not Y*
-- be able to not be able to X ==> not X
-- hesitate to not hesitate to X ==> X

She managed to get a job. She didn't manage to get a job.

He failed to get a job. He didn't fail to get a job.

She forced him to leave. She didn't force him to leave.

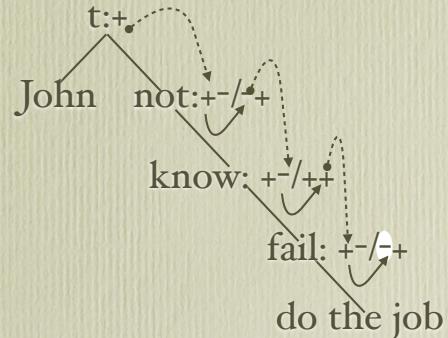
She prevented him from leaving. She didn't prevent him from leaving.

He wasn't able to leave.

He didn't hesitate to leave.

How this works

- John didn't know that Mary failed to do the job.



=> the job did not get done

Veridicality relations between contexts

- Generalized entailment: veridical
 - if c_2 is veridical with respect to c_1 , what is ‘true’ in c_2 is ‘true’ in c_1
- Inconsistency: antiveridical
 - if c_2 is antiveridical with respect to c_1 , what is ‘true’ in c_2 is ‘not true’ in c_1
- Consistency: averidical
 - if c_2 is averidical with respect to c_1 , what is ‘true’ in c_2 may or may not be ‘true’ in c_1

Instantiation of entities

- Existential commitments: depend on veridicality relations of contexts but also on the presuppositions of NPs

Instantiability assertions

- An instantiability assertion of a concept-denoting term in a context implies the existence of an instance of that concept in that context
- An uninstantiability assertion of a concept-denoting term in a context implies there is no instance of that concept in that context
- If the denoted concept is of type event, then existence/nonexistence corresponds to veridicality

Relation between veridicality and instantiability

- for events: as said above,
 - if an event is in a context that is veridical wrt to the higher context, then the event is instantiable in the higher context,
 - if an event is in a context that is antiveridical wrt the higher context that in is uninstantiable in the higher context
 - if an event is in a context that is averidical wrt the higher context then we do not know whether the event is instantiable in the higher context

- for entities
 - Even when they are in context that is antiveridical or averidical wrt the higher context, entities can be instantiable in the higher context, e.g. referents of proper names are instantiable in all contexts, in our system we also assume that definites are instantiable in all contexts

Lifting rules

How this is used

The system is used to detect the entailment or contradiction relation between two pieces of texts.

ECD: Entailment and contradiction detection

For two pieces of text, called the text and the hypothesis (following PASCAL's Recognising Textual Entailment Challenge), the system tries to calculate whether, the hypothesis follows from the text, whether it contradicts the text or whether it does neither.

To do this, the system calculates specificity relations and veridicity relations.

Entailment detection and instantiability

Any instantiable concept is entailed by an equally or more specific concept. (John hopped ==> John moved)

Any uninstantiable concept entails an equally specific or more specific uninstantiable concept
(John didn't hop <=> John didn't move)

Contradiction detection and instantiability

Contradiction arises with an event is instantiable in text a and uninstantiable in text b

Contradiction detection depends on instantiability

When an event is instantiable in text a and not instantiable in text b, we have a contradiction

No one moved

Conceptual structure:
subconcept(not:12,[not:1])
subconcept(one:o,[person:1])
subconcept(move:2,[travel-1,...])
role(theme,move:2,person:o,[person:1])
role(cardinality_restriction,one:o,no)

Contextual Structure:

context(t)
context(ctx(move:2))
top_context(t)
context_lifting_relation(antiveridical,t,ctx(move:2))
context_relation(t,ctx(move:2),not:12)
uninstantiable(move:2,t)
instantiable(move:2,ctx(move:2))
instantiable(one:o(ctx(move:2)))

Somebody moved

Conceptual structure:
subconcept(one:o,[person:1])
subconcept(move:2,[travel-1,...])
role(theme,move:2,person:o,[person:1])
role(cardinality_restriction,one:o,i)

Contextual Structure:

context(t)
top_context(t)
instantiable(move:2,t)
instantiable(one:o,t)

- In practice, the calculations are done by the elimination of the less specific information

Specificity computation and elimination

Alignment

Text:

Kim hopped.

Hypothesis:

Someone moved.

Specificity
computation

Text:

Kim hopped.

Hypothesis:

Someone moved.

Elimination of
H facts that are
entailed by T facts.

Text:

Kim hopped.

Hypothesis:

Someone moved.

Red: more specific; blue: less specific

Stages of the ECD

1. Alignment based on WordNet and Alias
2. See whether the roles of the aligned concepts are the same
3. If the roles in premise and the conclusion are not the same, the alignment is rejected.
4. Cardinality restrictions on aligned concepts are checked (including quantifiers, ...)
5. Any instantiable concept in the conclusion is removed if it is aligned with an equally or more specific concept in premise. (P: John hopped; C:John moved --> delete both John and moved from the conclusion)
6. Any uninstantiable concept in the conclusion is removed if it is equally specific or more specific than the concept in the premise it is aligned with.
7. Contradiction is checked (instantiable in premise and uninstantiable in conclusion, or vice versa)
8. If nothing left in conclusion, then conclusion follows