Paraphrasing controlled English texts

Kaarel Kaljurand

CNL 2009, Marettimo, Italy

2009-06-09

Outline

- What is a paraphrase?
- Usage and requirements
- Paraphrasing ACE by DRS verbalization
  - DRS → Core ACE
  - DRS → NP ACE
- Encountered problems, conclusions

Tool support for CNLs

- CNLs have formal syntax/semantics
  - just like programming languages
- thus enable various useful supporting tools
  - syntax highlighting, syntax error pinpointing, auto-completion, consistency checking, refactoring, etc., etc.
- A paraphraser is one of such tools

Definition

- A paraphrase of a text is its reformulation (in the same language) such that the meaning of the text is preserved.
  - Paraphrase cannot use meta-level such as color, font-size, full NL
  - We have to define what is meant by "meaning"
- Additionally, the text and its paraphrase should be syntactically different.
  - The language should contain syntactic sugar
- Example:
  - Mary is liked by everybody.
  - If there is somebody X then X likes Mary.

Possible uses

- Make the interpretation of the text more clear
  - point out constructs that are potentially misunderstood
- Reformulate the text so that it becomes easier to read
  - bring related sentences closer together
- Highlight constructs that are not supported in the underlying logic
  - e.g. the underlying DRS cannot be expressed in OWL
- …

Requirements

- Paraphrase should be different from the original (by definition)
  - How different? Similar sentence structure can help the user to better relate the paraphrase to the original.
- Mary is liked by John and **she** likes **him**.
  - Mary is liked by John and **Mary** likes **John**.
  - John likes Mary. Mary likes John.

Requirements

- Paraphrase language should be syntactically small
  - paraphrasing as "normalization" into a core subset of the full CNL
  - the (interpretation of the) core subset is probably easier to learn for the user

Requirements

- Paraphrase should improve readability
- Readability of a single sentence
  - Every book is a document that an author who a publisher likes writes.
    - Every book is a document that is written by an author who is liked by a publisher.
    - If there is a book X then X is a document and an author Y writes X and a publisher likes Y.
- Readability of the complete text
  - e.g. reorder sentences to avoid long-distance anaphoric references

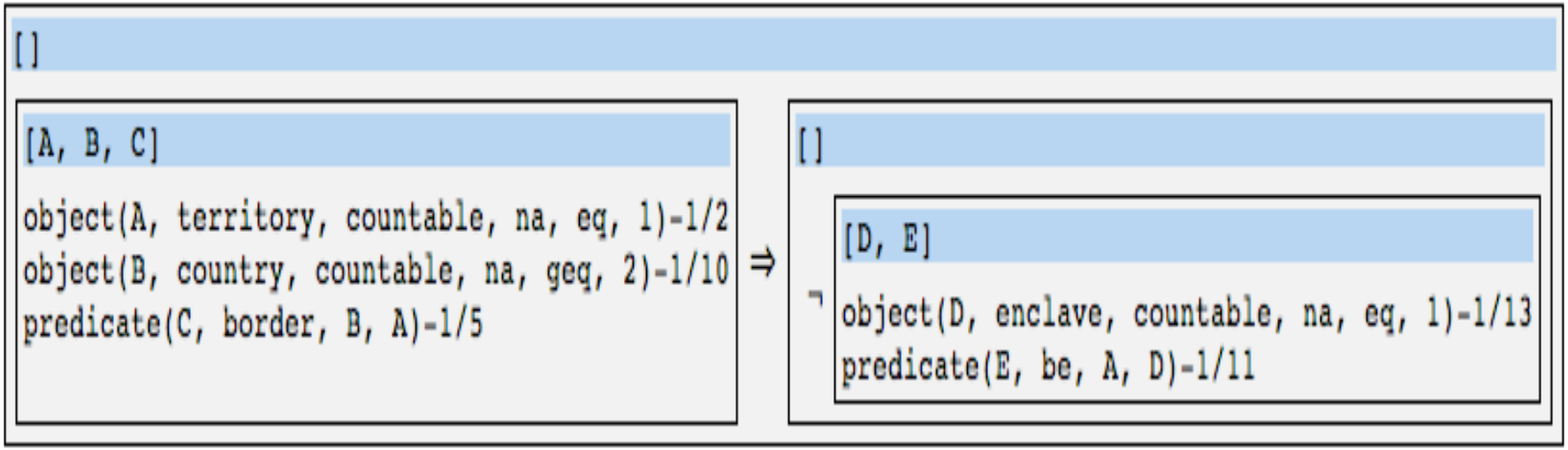
Requirements

- Paraphrase should teach the interpretation rules of the CNL
  - i.e. transform into a form that is less ambiguous in parent NL
- A dog is an animal.
  - There is a dog. The dog is an animal. (*a* is an existential quantifier)
- Every dog is an animal.
  - If there is a dog then the dog is an animal. (*every* corresponds to *if-then*)

Paraphrasing ACE texts

- Meaning of ACE texts given by the DRS
- DRS structural equivalence:
  - e.g. reordering DRS conditions is allowed
  - e.g. renaming variables and changing sentence/token IDs is allowed
  - e.g. removing double negation is not
- ACE provides syntactic sugar
  - various forms of coordination and negation, *every* vs *if-then*, *of* vs Saxon genitive, various forms of anaphoric references, sentence reordering
- Two paraphrase languages so far
  - Core ACE
  - NP ACE

DRS example



- No territory that is bordered by at least 2 countries is an enclave.
- If at least 2 countries border a territory X1 then it is false that the territory X1 is an enclave.

Core ACE: ideas

- Use the smallest syntactic subset of ACE (i.e. the core)
- "Flatten" the structure of sentences
  - remove relative clauses
  - split sentence conjunction into multiple sentences
- Fix the order of
  - sentences
  - elements in coordination
  - adjuncts (prepositional phrases and adverbs)

The Core ACE language

- Defined by removing some ACE constructs such that the semantic expressivity is not affected
  - quantifiers: *every*, *each*, *no*, *for each*, … (→ *if-then*)
  - passive (X is seen by Y → Y sees X)
  - Saxon genitive (John's dog → a dog of John)
  - VP negation
    - A man does not run. →
    - There is a man. It is false that the man runs.
  - relative clauses
    - Every man who loves a woman who loves him smiles. →
    - If a woman X1 loves a man X2 and the man X2 loves the woman X1 then the man X2 smiles.
  - pronouns
    - John sees somebody. He hates John's dog. →
    - John sees somebody X. X hates a dog of John.

NP ACE: ideas

- Conciseness (shorter sentences)
  - achieved by using relative clauses, instead of full clauses and explicit anaphoric references
- Focus only on implications (paraphrased as *every*-sentences)
  - support widespread rule and ontology language patterns
  - superset of the OWL verbalizer output language

The NP ACE language

- *If-then* sentences are represented as *every*-sentences
  - Boolean combinations of sentences are expressed by relative clauses
  - *if*-part and *then*-part must share arguments
  - Passive must be often used
- Cannot express all ACE constructs, missing:
  - NP pre-modifiers, VP modifiers, possessive constructs, ditransitive verbs, NP conjunction, numbers and strings, embedded *if-then* sentences
- No overlap with Core ACE

NP ACE: examples

- Argument sharing
  - If a man owns a dog then a woman owns a cat. →
  - *FAIL*
- Usage of passive
  - If a man owns **a car** then there is a woman who hates **the car**. →
  - Every **car** that is owned by a man is hated by a woman .

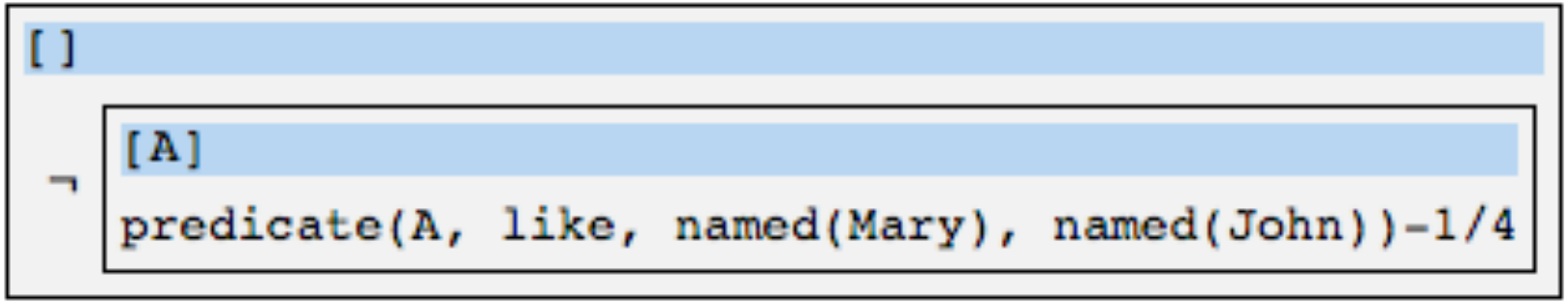
Implementation

- Paraphrase as a verbalization of the DRS of the input text
  - i.e. ACE1 → DRS1 → ACE2, where
  - ACE1 → DRS1 is an ACE parser
  - DRS1 → ACE2 is a DRS verbalizer
- Can automatically check if the paraphrase is correct, by ACE2 → DRS2, and checking DRS1 and DRS2 for structural equivalence

Core ACE verbalizer

- Applies a relatively direct transformation of DRS conditions into ACE sentences
  - *predicate*-conditions (i.e. conditions that correspond to verbs and their complements) map to simple ACE sentences
  - embedded DRSs map to complex sentences (e.g. negated or *if-then*-sentences)
  - content word lemmas are mapped to surface forms using the same lexicon that was used to obtain the DRS
- The order of sentences that originate from the same DRS is fixed so that sentences that mention the same nouns are positioned next to each other (in the conjunction).
  - This will result in easier to read sentences.

Example



- It is false that Mary likes John.

Core ACE verbalizer coverage

- Tested on APE regression test set (2421 ACE→DRS mappings)
- 88% correctly paraphrased
- 9% of the paraphrases identical to the original
- Not covered
  - *each of* plurals
  - complex forms of questions
  - …

NP ACE verbalizer

- Only applied to DRS implications which furthermore must share at least one discourse referent between the *if*-box and the *then*-box.
  - Only such implications can be expressed as *every*-sentences.
- The *predicate*-conditions in both the *if*-box and the *then*-box are "rolled up" starting with the condition that contains a shared discourse referent.
- The resulting structures are directly mapped to noun phrases that are possibly modified by (a coordination or negation of) relative clauses.

Problems

- Paraphrase sometimes identical to the original
  - Examples
    - John likes Mary.
    - Every airline charges a passenger with an overweight-luggage.
  - Solution: use other means of explanation
- Handling complex scopes
  - {Every dog is an animal} or {there is a cat}.
  - If there is a dog X1 then {{the dog X1 is an animal} or {there is a cat}}.

Availability

- Two DRS verbalizers (into Core ACE and into NP ACE) are included with the Attempto Parsing Engine (APE)
  - http://attempto.ifi.uzh.ch/site/downloads/

Conclusions

- Two non-overlapping fragments, often offering two alternative formulations of the original text
- Useful form of feedback for the user
  - simplifies complex structures
  - teaches interpretation rules
  - useful for DRS checking (for an ACE parser developer)

Thank You!