How to Evaluate Controlled Natural Languages

Tobias Kuhn

Workshop on Controlled Natural Language (CNL 2009), Marettimo, Italy

8 June 2009
Off Topic: AceWiki

- We use here the definition of "planet" according to the International Astronomical Union (see http://www.iau.org) without the restriction to solar planets.
- Every planet is a celestial-body.
- No planet is a star.
- No planet is a dwarf-planet.
- The distinction between planet and dwarf-planet has been introduced by the International Astronomical Union in 2006.
- No planet is a moon.
- Every planet orbits a star.
- Every planet that does not orbit the Sun is an extrasolar planet.
- Which planets orbit the Sun?
  - Earth
  - Jupiter
  - Mars
  - Mercury
  - Neptune
  - Saturn
  - Uranus
  - Venus
- Every planet is a terrestrial planet or is a gas giant.
Of Topic: ACE Editor

ACE Editor  File  Edit  View

> John is an important customer.

ACE Text Editor

Every important customer can be

< Delete

< text

function word

a  an  as  every  everybody  everything  more

proper name

Austria  Berlin  Bill  Canada  France  Germany  Italy

adjective

active  angrier  angriest  angry  authenticated  automatic  average

transitive adjective

fond-of  fonder-of  interested-in  located-in  mad-about  madder-about  registered-at

new variable

X  Y  Z  X1  Y1  Z1  X2

reference

the customer

passive verb

accepted  assigned  awaited  beaten  believed  bitten  blamed

OK  Cancel
Introduction

- (Formal) Controlled Natural Languages (CNL) are designed to be more understandable and more usable by humans than common formal languages.
- But how do we know whether this goal is achieved?
- The only way to find out: User Studies!
Evaluation of CNL Tools

- Many user studies have been performed to evaluate tools that use CNL, e.g. [1].

- Hard to determine how much the CNL contributes to the understandability

- Hard to compare CNLs to other formal languages because different languages usually require different tools

Tool-Independent Evaluation of CNLs

- Only very few evaluations have been performed that test a CNL independently of a particular tool.

- [2] presents a paraphrase-based approach: The subjects of an experiment receive a CNL statement and have to choose from four paraphrases in natural English:

  Bob is an instance of an acornfly.
  - Bob is a unique thing that is classified as an acornfly.
  - Bob is sometimes an acornfly.
  - All Bob's are types of acornflies.
  - All acornflies are examples of Bob.
  - Unsure

Challenges with Paraphrase-based Approaches

- Ambiguity of natural language
  - One has to make sure that the subjects understand the natural language paraphrases in the right way.

- Does good performance imply understanding?
  - The formal statement and the paraphrases tend to look very similar if both rely on English.
  - One has to exclude that the subjects do the right thing without understanding the statements:
    - Following some syntactic patterns
    - Misunderstanding both – statement and paraphrase – in the same way
My Approach: Ontograph Framework

- Using a simple graphical notation: Ontographs
  - Designed to be used in experiments
  - Idea: Let the subjects perform tasks on the basis of situations depicted by diagrams (i.e. Ontographs).

- Assumption: Ontographs are very easy to understand.

- Every present is bought by John.
- John buys at most one present.
Ontographs consist of a **legend** and a **mini world**.

The legend introduces types and relations.

The mini world shows the existing individuals, their types, and their relations.
Ontographs: Properties

- Formal language
- Intuitive graphical icons
- No partial knowledge
- No explicit negation
- No generalization
- Large syntactical distance to textual languages
The goal of the experiment was to find out whether controlled natural languages are more understandable than comparable common formal languages.

- **CNL**: Attempto Controlled English (ACE)
- **Comparable language**: Manchester OWL Syntax [3]:
  
  »The syntax, which is known as the Manchester OWL Syntax, was developed in response to a demand from a wide range of users, who do not have a Description Logic background, for a “less logician like” syntax. The Manchester OWL Syntax is derived from the OWL Abstract Syntax, but is less verbose and minimises the use of brackets. This means that it is quick and easy to read and write.«

- **For a direct comparison, we defined a slightly modified version**: MLL (Manchester-like language)

ACE versus MLL

Bill is not a golfer.

Bill $\text{HasType not golfer}$

No golfer is a woman.

golfer $\text{DisjointWith woman}$

Nobody who is a man or who is a golfer is an officer and is a traveler.

man or golfer $\text{SubTypeOf not (officer and traveler)}$

Every man buys a present.

man $\text{SubTypeOf buys some present}$

Lisa helps at most 1 person.

Lisa $\text{HasType helps max 1 person}$

If X helps Y then Y does not love X.

helps $\text{DisjointWith inverse loves}$
Learning Time

understanding

controlled natural language
common formal language

learning time

0 20 min 4 h 2 weeks 1 year
4 Series of Ontographs
## Statements in ACE and MLL for each Ontograph

<table>
<thead>
<tr>
<th>Ontograph</th>
<th>Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a- People</td>
<td>Everything that sees something is an officer.</td>
</tr>
<tr>
<td>1b+ Loves</td>
<td>Everything that loves something is a person.</td>
</tr>
<tr>
<td>2a- Loves</td>
<td>Everything that is loved by something is a person.</td>
</tr>
<tr>
<td>2b+ Loves</td>
<td>Everything that is bought by something is a present.</td>
</tr>
<tr>
<td>3a- Loves</td>
<td>Everything that loves something is a traveler or an officer.</td>
</tr>
<tr>
<td>3b+ Loves</td>
<td>Everything that sees something is an officer or is a traveler.</td>
</tr>
<tr>
<td>4a+ Sees</td>
<td>Everything that is seen by something is a traveler or is an aquarium.</td>
</tr>
<tr>
<td>4b- Sees</td>
<td>Everything that is bought by something is an aquarium or is an officer.</td>
</tr>
<tr>
<td>5a+ Loves</td>
<td>Tom loves at least 2 officers.</td>
</tr>
<tr>
<td>5b- Sees</td>
<td>Sue sees at least 2 persons.</td>
</tr>
<tr>
<td>6a- Loves</td>
<td>Lisa buys at most 1 present.</td>
</tr>
<tr>
<td>6b+ Loves</td>
<td>Bill loves at most 1 person.</td>
</tr>
<tr>
<td>7a+ Sees</td>
<td>Every traveler sees at least 2 aquariums.</td>
</tr>
<tr>
<td>7b- Sees</td>
<td>Every officer buys at least 2 presents.</td>
</tr>
<tr>
<td>8a+ Sees</td>
<td>Everything that buys at least 2 presents is an officer.</td>
</tr>
<tr>
<td>8b- Sees</td>
<td>Everything that loves at least 2 presents is an officer.</td>
</tr>
<tr>
<td>9a+ Sees</td>
<td>Every officer sees at most 1 aquarium.</td>
</tr>
<tr>
<td>9b- Sees</td>
<td>Every person buys at most 1 present.</td>
</tr>
<tr>
<td>10a- Loves</td>
<td>Everything that is a traveler or that is an officer sees at most 1 aquarium.</td>
</tr>
<tr>
<td>10b+ Loves</td>
<td>Everything that is an officer or that is a traveler loves at most 1 person.</td>
</tr>
</tbody>
</table>
Experiment: Subjects

- Requirements:
  - Students, but no computer scientists or logicians
  - At least intermediate level in written German and English

- Recruitment of 64 subjects:
  - Broad variety of fields of study
  - On average 22 years old
  - 42% female, 58% male

- The subject were equally distributed into eight groups:
  (Series 1, Series 2, Series 3, Series 4) x (ACE first, MLL first)
Experiment: Procedure

1. Subjects read an instruction sheet that explains the procedure, the pay-out, and the ontograph notation.

2. The subjects answer control questions in order to check whether they understood the instructions.

3. During a learning phase that lasts at most 16 minutes, the subjects read a language description sheet (of either ACE or MLL) and see on the screen an ontograph together with 10 statements marked as “true” and 10 marked as “false”.

4. During the test phase that lasts at most 6 minutes, the subjects see another ontograph on the screen and have to classify 10 statements as “true”, “false”, or “don't know”.

5. The steps 3 and 4 are repeated with the other language.

6. The subjects fill out a questionnaire.
Language Instruction Sheets: ACE versus MLL

Sprache A

Die Sprache A basiert auf Aussagen in englisch mit bestimmten Interpretationen in Hinblick auf die Bedeutung der Benennung der Verben und die Verben entsprechen den Determinanten in der Sprache. Die Substantive entsprechen den Typen und die Verben entsprechen den Prädikaten. Im folgenden werden nun die Interpretationen der Aussagen erklärt.

„something“/„everybody“/„nothing“

Die Wörter "something", "everybody", und "nothing" können sich immer auf jemanden beziehen. Normalerweise bedeutet "something" im Englischen nichts, was es auf eine Person bezieht. Dies führt zu einer Interpretation der Aussage "John loves something" als "John likes something" in Sprache A.

"nothing but"

Die Wörter "nothing but" werden in der Kombination "nothing but" verwendet, was "nichts anderes" bedeutet. "John sees nothing but" bedeutet, dass John nichts anderes sieht als Menschen, die ihm zur Verfügung stehen.

Inhaltliche Interpretation

Interpretationen sollten nicht bei der Ableitung, sondern in der Sprache A richtig oder falsch sein, basierend auf die Interpretation abstrahiert, die man als englischsprachige Person intuitiv aus der Aussage herausleitet.

Sprache B

Die Sprache B besteht aus Aussagen, wie sie unten beschrieben sind. Diese Aussagen sind aus Schlüsselwörtern und den Namen der individuellen Typen und Determinanten, die entsprechenden Hilfsfunktionen aufgebaut. Die verwendeten Schlüsselwörter sind "desiTypte", "SubTypete", "mit", "same" und "only".

Aussagen


Positiver einfacher Aussage

Schema: Individuelles Namensheits-Typ
typ (mit)

Beispiel: John sees Mary

Erklärung: Eine positive einfache Aussage besteht aus zwei Individuen, von denen eines (Mary) das andere Individuum (John) beobachtet (sich anspricht).

SubTypete-Aussage

Schema: SubTypete

Beispiel: Peter SubTypete

Erklärung: Eine SubTypete-Aussage verleiht ein Individuum einen SubTyp (Peter SubTypete).

Negativer einfacher Aussage

Schema: Negativer einfachen Namensheits-Typ (mit)

Beispiel: Mary not seen by Jack

Erklärung: Eine negative einfache Aussage besteht aus zwei Individuen, von denen eines (Jack) das andere Individuum (Mary) nicht beobachtet (sich anspricht).

Typ-Operationen

Jede Typ-Funktion ist komplexe für eine ganze Gruppe von Individuen. Eine typische Funktion ist zum Beispiel "same" oder "different". Bei der Verwendung der Typ-Operationen hängt die Funktion "same" im Beispiel "same" von der Position ab.

sub-set-Operator

Schema: sub-set

Beispiel: John only woman

Erklärung: Der sub-set-Operator verleiht nur einen Typ, die hier sub-set ist, der nicht zum gegebenen Typ gehört. Der sub-set Operator stellt für eine Individuen, die keine Anzahl enthält.

same-Operator

Schema: same-type

Beispiel: Mary same-type

Erklärung: Der same-Operator verleiht eine Funktion und einen Typ. Der allgemeinste Typ "same" entsteht, wenn eine "same" für alle Individuen in einer bestimmten Gruppe gehalten wird. Der sub-set-Operator genügt dies nicht, um alle Individuen in einer bestimmten Gruppe zu identifizieren.
Experiment: Learning Phase

True statements:
- Mary not sees Tom
- Mary HasType not (sees some man)
- John HasType buys some (not present)
- John sees Tom
- Tom HasType sees only woman
- buys some present SubTypeOf man
- woman SubTypeOf buys only picture
- buys only picture SubTypeOf woman
- John HasType buys some present
- man SubTypeOf buys some present

False statements:
- John HasType sees only man
- man SubTypeOf buys only present
- sees some woman SubTypeOf man
- Tom HasType buys some picture
- John HasType not (sees some woman)
- Tom not sees Lisa
- woman SubTypeOf buys some picture
- sees only woman SubTypeOf man
- Tom HasType sees some (not woman)
- Lisa sees Mary

Time left for this page: 15:42
Experiment: Testing Phase

Which of the statements are true and which are false?

- Sue HasType admires some (not traveler)
- traveler SubTypeOf inspects some letter
- Tom HasType inspects only letter
- Lisa not admires Tom
- officer SubTypeOf inspects only aquarium
- Tom inspects Lisa
- admires only person SubTypeOf officer
- Inspects some aquarium SubTypeOf officer
- Paul HasType not (admires some officer)
- Lisa HasType admires some officer

Mini World

Legend

person
traveler
officer
aquarium
letter
inspects
admires

Time left for this page: 5:39
Experiment: Pay-out

- Every subject got 20.00 CHF for participation.
- Furthermore, they got 0.60 CHF for every correctly classified statement and 0.30 CHF for every “don't know”.
- Thus, every subject earned between 20 and 32 CHF.
Evaluation: Ontograph Framework

- Did the Ontograph framework work? Answer: Yes!
  - The subjects performed very well in the experiment (8.9 correct classifications out of 10)
  - They found the ontographs very easy to understand (questionnaire score of 2.7 where 0 is “very hard to understand” and 3 is “very easy to understand”)

![Correct Classifications per Test Phase](chart1)

![Ontograph Understandability](chart2)
Evaluation: ACE vs MLL

- Which language performed better?
- Answer: ACE was understood better, within shorter time, and was liked better by the subjects than MLL!

\begin{itemize}
  \item \textbf{ACE} vs \textbf{MLL}
  \item \textbf{Correct Classifications per Test Phase}
  \item \textbf{Time in Minutes for Learn and Test Phase}
  \item \textbf{Questionnaire Score for Understandability}
\end{itemize}

\textit{p-values obtained by Wilcoxon signed rank test:}

- 0.003421
- 1.493e-10
- 3.24e-07
Evaluation: First/Second Language

- Correct Classifications per Test Phase
- Time in Minutes for Learn and Test Phase
- Questionnaire Score for Understandability
Evaluation: Series 1/2/3/4

![Bar charts showing correct classifications per test phase and questionnaire score for understandability for Series 1 to Series 4.]
Evaluation: Regression

Regression on the 128 test phase results with the normalized classification score (-5 to 5) as the dependent variable.

Baseline: testing MLL as second language on series 1, male subject of 18 years with good (but not very good) English skills.

|               | Robust sc_norm |      Coef.   | Std. Err.   |     t    | P>|t|    |
|---------------|----------------|--------------|-------------|---------|--------|
| ace           | .5156250       | .1800104     | 2.86        | 0.006   |
| first_lang    | -.2187500      | .1800104     | -1.22       | 0.229   |
| series_2      | -.4802784      | .3371105     | -1.42       | 0.159   |
| series_3      | -.2776878      | .3485605     | -0.80       | 0.429   |
| series_4      | -.8795029      | .5219091     | -1.69       | 0.097   |
| female        | .1413201       | .2982032     | 0.47        | 0.637   |
| age_above_18  | -.0724091      | .0296851     | -2.44       | 0.018   |
| very_good_engl| .2031366       | .2967447     | 0.68        | 0.496   |
| _cons         | 4.302329       | .3251371     | 13.23       | 0.000   |
Conclusions

- The Ontograph framework seems to be suitable for understandability experiments for CNLs.
- **ACE is understood significantly better than MLL.**
  - There is no reason to believe that another logic syntax (except CNLs) would have performed better than MLL.
- Furthermore, ACE requires significantly less time to be learned and was liked better by the subjects.
Resources for the Ontograph Framework

- The resources for the Ontograph framework are available freely under a Creative Commons license:
- [http://attempto.ifi.uzh.ch/site/docs/ontograph/](http://attempto.ifi.uzh.ch/site/docs/ontograph/)
Thank you for your attention!

Questions/Discussion