Logical analysis of natural language

Marek Medveď

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno

xmedved1@fi.muni.cz

PA153 Natural Language Processing

Outline

1 Motivation

2 Predicate logic

3 Transparent intensional logic

4 Propositional logic

Logical analysis of natural language

- Sentence → logical formula
  - formal reasoning
  - interlingua for machine translation
  - precise expressions
- Which formalism?
  - (first order) predicate logic
  - propositional logic
  - modal logics
  - intensional logics (IL, Richard Montague)
  - transparent intensional logic (TIL)

Natural language → predicate logic

- What are formulas for
  - „Some prime numbers are even"
  - „Some odd numbers are even"
  - „Some smart people are lazy"
  - „No bachelor is married"
  - „No bachelor is rich"
  - „Miloš Zeman is the president of CR."
  - „Karel counts 5 + 7"
- What is wrong?

Natural language → predicate logic

- What are formulas for
  - „Some prime numbers are even"
  - „Some odd numbers are even"
  - „Some smart people are lazy"
  - „No bachelor is married"
  - „No bachelor is rich"
  - „Miloš Zeman is the president of CR."
  - „Karel counts 5 + 7"
- What is wrong?
  - different types of truth
  - granularity of the description is insufficient

Natural language → predicate logic

Karel counts $5 + 7$
$5 + 7 = 12$

Karel counts 12

Miloš Zeman is the president of CR.
Karel Schwarzenberg wanted to become the president of CR.

Karel Schwarzenberg wanted to become Miloš Zeman.

It is not true that the king of France is bald-headed

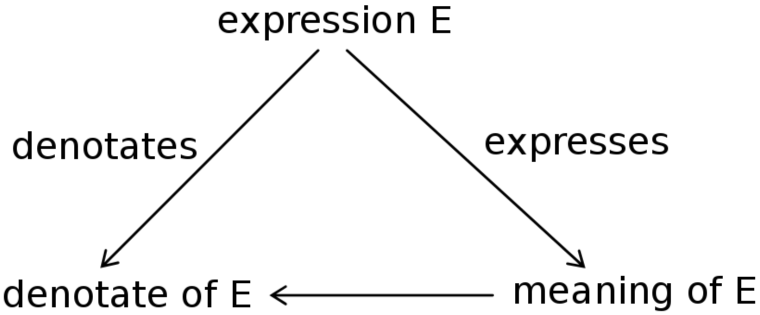
The king of France has some hair on his head.

Natural language $\longrightarrow$ predicate logic

- Predicate logic can be used for natural language analysis
  - some applications are doing it
  - but not in all cases
  - we need to be aware of the limits
  - work-arounds are possible but may be complicated
- Advantages of predicate logic
  - it is simple
  - it is well explored
  - inference machine exists

What is the meaning of an expression?

Frege's model of semantics



Conception of possible worlds

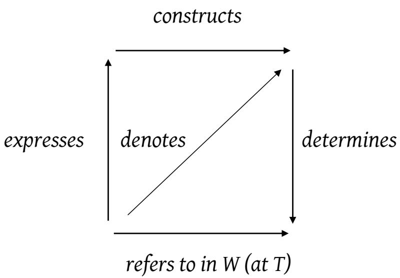
- A possible world
    - a set of non-contradictory formulas about the universe
    - the current world is one of the possible worlds
- Empirical truth
    - the truth of a formula depends on the particular world
    - meaning is always world-independent
- Intensional logics
    - intensions (world-independent)
    - extensions (denotates, objects in a particular world)

Transparent intensional logic

- Pavel Tichý, Pavel Materna
- Procedural logic
  - possible worlds + possible times
  - meaning is a emphconstruction, i.e. abstact procedure (algorithm) which takes the current world and time and outputs the denotate (the particular object)
  - coded as lambda function
- Typed logic
  - o – true, false
  - ι – set of individuums
  - τ – set of real numbers (or time moments)
  - ω – set of possible worlds
  - $((o\tau)\omega)$ – proposition
  - $(((o\iota)\tau)\omega)$ – property $((o\iota)_{\tau\omega})$

Tichý model of semantics

meaning ***M*** (construction) denotatum ***D*** (intension/extension)



expression ***E*** referent ***R*** (the value of an intension in *W*)

Transparent intensional logic – examples

Miloš Zeman je prezidentem CR.

$\lambda w \lambda t$[= Miloš_Zeman President_CR$_{wt}$]

type $o_{\tau\omega}$ Miloš_Zeman/ι President_CR/$\iota_{\tau\omega}$ $= /(o\iota\iota)$

Schwarzenberg wanted to become CR.

$\lambda w \lambda t$[want_to_become$_{wt}$ Schwarzenberg President_CR]

$o_{\tau\omega}$ Schwarzenberg/ι President_CR/$\iota_{\tau\omega}$ want_to_become/$(o\iota\iota_{\tau\omega}$

Transparent intensional logic – examples

5 + 7

$[\,+\;5\;7\,]$

type τ $\quad 5,7/\tau \quad +/(\tau\tau\tau)$

Karel counts 5 + 7.

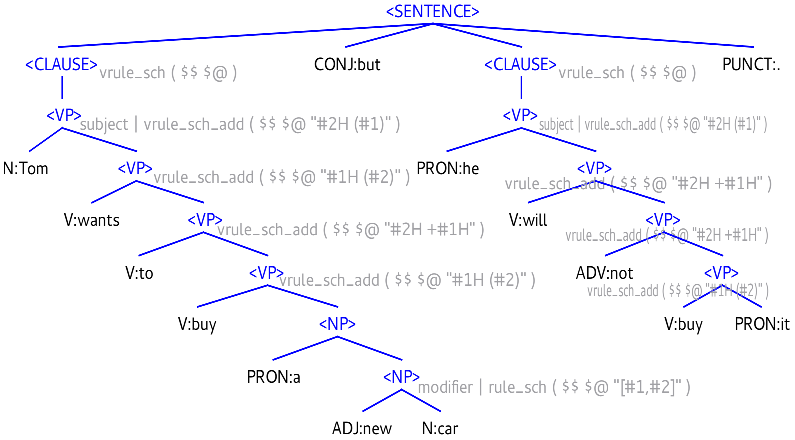
$\lambda w \lambda t[\,\mathsf{count}_{wt} \quad \mathsf{Karel} \quad {}^{0}[\,+\;5\;7\,]\,]$

type $o_{\tau\omega}$ $\quad {}^{0}[+\;5\;7]/{}^{*}1 \quad \mathsf{count}/(o\iota{*}_1)_{wt} \quad \mathit{Karel}/\iota$

Normal translation algorithm

- Automatic conversion of sentences into TIL constructions
    - doc. Aleš Horák
    - morphological analysis
    - syntactic analysis
    - conversion from tree to TIL formula
    - type check
    - output of all the layers is ambiguous
    - implementation within the Synt parser, currently experiments with other parsers (SET)
- Further requirements
    - lexicon of types ("snow" vs. "give" – exploitation of valency lexicons)
    - rules for type control

Example


<SENTENCE>
<CLAUSE> vrule_sch ($$ $@)
CONJ:but
<CLAUSE> vrule_sch ($$ $@)
PUNCT:.
<VP> subject | vrule_sch_add ($$ $@ "#2H (#1)")
N:Tom
<VP> vrule_sch_add ($$ $@ "#1H (#2)")
V:wants
<VP> vrule_sch_add ($$ $@ "#2H +#1H")
V:to
<VP> vrule_sch_add ($$ $@ "#1H (#2)")
V:buy
<NP>
PRON:a
<NP> modifier | rule_sch ($$ $@ "[#1,#2]")
ADJ:new
N:car
<VP> subject | vrule_sch_add ($$ $@ "#2H (#1)")
PRON:he
<VP> vrule_sch_add ($$ $@ "#2H +#1H")
V:will
<VP> vrule_sch_add ($$ $@ "#2H +#1H")
ADV:not
<VP> vrule_sch_add ($$ $@ "#1H (#2)")
V:buy
PRON:it

Example

$$\lambda w_1 \lambda t_2 \Big((\exists x_3)(\exists i_4)(\exists i_5) \Big(\big[\mathbf{Does}_{w_1 t_2}, i_5, [\mathbf{Imp}_{w_1}, x_3]\big] \wedge \big[[\mathbf{new}, \mathbf{car}]_{w_1 t_2}, i_4\big]$$
$$\wedge\ x_3 = [\mathbf{to\_want}, i_4]_{w_1} \wedge \big[\mathbf{Tom}_{w_1 t_2}, i_5\big]\Big) \wedge \Big[\mathbf{Not}, \Big[\mathbf{True}_{w_1 t_2},$$
$$\lambda w_6 \lambda t_7 (\exists x_8)(\exists i_9) \Big(\big[\mathbf{Does}_{w_6 t_7}, He, [\mathbf{Perf}_{w_6}, x_8]\big] \wedge \big[\mathbf{it}_{w_6 t_7}, i_9\big]$$
$$\wedge\ x_8 = [\mathbf{to\_buy}, i_9]_{w_6} \Big)\Big]\Big]\Big) \ldots \pi$$

Transparent intensional logic – pros and cons

- Advantages
  - correct and very precise analysis
  - makes general correct reasoning possible
- Disadvantages
  - very abstract and complex
  - not really wide-spread
  - experts often do not agree on correct analysis

Transparent Intensional logic on FI

- doc. Aleš Horák
- prof. Marie Duží
    - subjects Introduction to Transparent Intensional Logic
- Small corpus of correct constructions for Czech
    - `https://corpora.fi.muni.cz/til`
- Semantic network of constructions
    - as a knowledge base for automatic reasoning

Propositional logic

- used in logic programming in for of Horn clauses

  T <- M

- can be used in Inductive Concept learning

Inductive concept learning

- aims to learn an **intensional description** by induction from positive, negative examples and background knowledge
- induced description is also called hypothesis

Inductive concept learning

$$\mathcal{E}^+ = \begin{cases} transport\_by\_land(bike). \\ transport\_by\_land(motorbike). \\ transport\_by\_land(car). \\ transport\_by\_land(jeep). \\ transport\_by\_land(truck). \\ transport\_by\_land(bus). \\ transport\_by\_land(hovercraft). \end{cases}$$

$$\mathcal{E}^- = \begin{cases} transport\_by\_land(airplane). \\ transport\_by\_land(seaplane). \\ transport\_by\_land(airship). \\ transport\_by\_land(helicopter). \end{cases}$$

$$\mathcal{B} = \begin{cases} has\_propeller(hovercraft). \\ has\_propeller(airplane). \\ has\_propeller(seaplane). \\ has\_propeller(helicopter). \\ has\_propeller(airship). \\ \\ has\_steering\_wheel(car). \\ has\_steering\_wheel(truck). \\ has\_steering\_wheel(bus). \\ has\_steering\_wheel(jeep). \\ \\ travels\_on\_wheels(motorbike). \\ travels\_on\_wheels(bike). \\ \\ vertical\_take\_off(helicopter). \\ vertical\_take\_off(airship). \\ \\ has\_wings(airplane). \\ has\_wings(seaplane). \\ \\ travels\_on\_wheels(X) \leftarrow has\_steering\_wheel(X). \end{cases}$$

Inductive concept learning

$$\mathcal{H}^{\mathcal{E}}_{\mathcal{B}} = \begin{cases} transport\_by\_land(X) \leftarrow travels\_on\_wheels(X). \\ transport\_by\_land(hovercraft). \end{cases}$$

Detecting outliers through concept learning

$$\mathcal{H}_{\mathcal{B}}^{\mathcal{E}} = \left\{ \begin{array}{l} \mathit{transport\_by\_land}(X) \leftarrow \mathit{travels\_on\_wheels}(X). \\ \mathit{transport\_by\_land}(\mathit{hovercraft}). \end{array} \right.$$

$$\mathcal{H}_{\mathcal{B}}^{\mathcal{E}\setminus\mathcal{O}} = \left\{ \mathit{transport\_by\_land}(X) \leftarrow \mathit{travels\_on\_wheels}(X). \right.$$

$$\mathcal{H}_{\mathcal{B}}^{\overline{\mathcal{E}}} = \left\{ \begin{array}{l} \mathit{not\_transport\_by\_land}(X) \leftarrow \mathit{has\_wings}(X). \\ \mathit{not\_transport\_by\_land}(X) \leftarrow \mathit{vertical\_take\_off}(X). \end{array} \right.$$

$$\mathcal{H}_{\mathcal{B}}^{\overline{\mathcal{E}}\setminus\overline{\mathcal{O}}} = \left\{ \mathit{not\_transport\_by\_land}(X) \leftarrow \mathit{has\_propeller}(X). \right.$$

- Exploiting domain knowledge to detect outliers (Fabrizio Angiulli · Fabio Fassetti)