An overview of applications of logic to computational linguistics

From Leibniz, Ongoing Passion for Life, Math, and Computers

Roussanka Loukanova

Logics for Linguistics

22 October, 2014

Languages & Information Flow

Major ways of using natural and some artificial languages:

- for information exchange between information processing systems involving
  - human communications
  - human–machine communications and interactions
  - bio-systems
- as reasoning media, for example, in
  - human cognition
  - computations (by humans and devices)
  - automated (i.e., computerized) reasoning systems, such as automatic provers and model checkers.

♢ In all these cases, language units are (typically) associated with denotations.

Computational linguistics covers various subjects

- *Computational Phonology* and *Computational Morphology*
- *Formal Syntax*; *Computational Syntax*
- *Formal Semantics*; *Computational Semantics*
- *Formal Grammar*; *Computational Grammar*
- *Parsing*

Focusing on syntactic phenomena

Often (but not always), these subjects develop methods, techniques, and theories for defining language fragments, by focusing on syntactic phenomena.

Well-formedness of syntactic structures

The methods and the algorithms associated with the above subjects are predominantly concerned with syntactic structures of (well-formed) words and expressions.

Some questions

- What is formal syntax?
  - formal definitions of the set of all, and only, well-formed expressions in a language, or a collection of languages
  - How? Via co-occurrence constraints expressed by grammar rules, principles, etc. (depending on the syntactic theory).
  - How are the co-occurrence rules expressed?
    By definitions of what are the well-formed constituent structures, usually trees
  - providing grammatical information associated with the well-formed tree structures
- Why language theories concentrate on constituent structures (usually trees), not just well-formed sequences of words?
- What is computational syntax?

To answer the above questions, in a better way, let us look at what is semantics (formal and computational)?

- Semantics is the study of the relations between
  - language units
    and
  - objects (material or abstract, atomic or complex) designated by those language units

Immediate questions:

- What are the language units?
- What are the objects designated by language units?

- What are the language units, and how they are defined, vary depending on the language and chosen syntactic theory of it
  - lexical items, words, phrases, sentences and/or utterances
  - syntactic categories, e.g., via
    - feature structure descriptions using feature-value pairs,
    - dependent types
  - constituent structures (usually trees, or tree-like diagrams)
- The kinds of objects designated by the language units vary depending on the language and chosen semantic theory of it.
  - denotations
  - intensions
- The notions of denotation and intension are formalized by using logic languages and theories
  - First Order Languages and Logic
  - Higher Order Languages and Logic / Type Theories
  - Information theories — model theories

- What are the language units, and how they are defined, vary depending on the language and chosen syntactic theory of it
  - lexical items, words, phrases, sentences and/or utterances
  - syntactic categories, e.g., via
    - feature structure descriptions using feature-value pairs,
    - dependent types
  - constituent structures (usually trees, or tree-like diagrams)
- The kinds of objects designated by the language units vary depending on the language and chosen semantic theory of it.
  - denotations
  - intensions
- The notions of denotation and intension are formalized by using logic languages and theories
  - First Order Languages and Logic
  - Higher Order Languages and Logic / Type Theories
  - Information theories — model theories

- What are the language units, and how they are defined, vary depending on the language and chosen syntactic theory of it
  - lexical items, words, phrases, sentences and/or utterances
  - syntactic categories, e.g., via
    - feature structure descriptions using feature-value pairs,
    - dependent types
  - constituent structures (usually trees, or tree-like diagrams)
- The kinds of objects designated by the language units vary depending on the language and chosen semantic theory of it.
  - denotations
  - intensions
- The notions of denotation and intension are formalized by using logic languages and theories
  - First Order Languages and Logic
  - Higher Order Languages and Logic / Type Theories
  - Information theories — model theories

Computational Syntax: Well-formedness (Artificial L, NL, HL)

- rigorously defined syntax
- algorithms for computing well-formed constituent structures

Computational Semantics: theories, methods, techniques, algorithms for

- defining denotations designated by well-formed structures
- defining adequate notion of sense
- models of information

∴ Syntax-Semantics Interface

Computational semantics and computational syntax bind together to give rise of computational theory of language (as comprehensive as possible), by interleaving syntactic and semantic phenomena.

Computational Syntax: Well-formedness (Artificial L, NL, HL)

- rigorously defined syntax
- algorithms for computing well-formed constituent structures

Computational Semantics: theories, methods, techniques, algorithms for

- defining denotations designated by well-formed structures
- defining adequate notion of sense
- models of information

∴ Syntax-Semantics Interface

Computational semantics and computational syntax bind together to give rise of computational theory of language (as comprehensive as possible), by interleaving syntactic and semantic phenomena.

Computational Syntax: Well-formedness (Artificial L, NL, HL)

- rigorously defined syntax
- algorithms for computing well-formed constituent structures

Computational Semantics: theories, methods, techniques, algorithms for

- defining denotations designated by well-formed structures
- defining adequate notion of sense
- models of information

∴ Syntax-Semantics Interface

Computational semantics and computational syntax bind together to give rise of computational theory of language (as comprehensive as possible), by interleaving syntactic and semantic phenomena.

Computational semantics

Computational semantics is a special case of information theory. Information theory covers wide range of topics:

- The concepts of information, information representation, information transfer
- Communication
  - *human* ↔ *human*
  - *human* ↔ *computer system*
  - inside and between bio-systems
- Reasoning:
  human cognition, *automated reasoning systems*

Computational semantics

Computational semantics combines and uses results, methods and techniques from from several fields:

- *Formal Semantics*: a subfield of math logic
- Theory of *Computability*: a subfield of the theoretical foundations of Computer Science (e.g., an interdisciplinary branch between math logic and Computer Science)
- Computer Science: concepts of programming, algorithms, data structures, etc.
- Theoretical Linguistics
- Computational Linguistics: formal and computational grammar, phonology, morphology, syntax
- Speech processing

Computational Grammar as an interdisciplinary field

- Requires Computational Syntax
- Requires Computational Semantics
  - Semantics has been inspired by philosophy of language, mind and reasoning.
- Requires Computational Lexicon
- Computational grammar accumulates scientific results from Mathematics, Mathematical Logic, Linguistics, Computer Science.
- Computational grammar is a subfield of interdisciplinary areas such as Artificial Intelligence and Cognitive Science.
- In recent years, computational grammar relates to Neuroscience, bio-informatics, etc., life sciences.
- The target is a computational, and adequate, grammar theory covering lexica, syntax, and semantics

Overview of approaches to formal syntax

CFGs, Phrase Structure Grammars: initiated by Chomsky 1950s

Transformational Grammars: initiated by Chomsky 1955, 1957,to the present

Generative Semantics: 1967-74; Lakoff, McCawley, Postal, Ross

GBT: initiated by Chomsky 1981

Minimalist Program; Principles and Parameters initiated by Chomsky 1981

Constraint-Based, Lexicalist Approaches

- *GPSG:* Gazdar et al. 1979-87 to the present
- *LFG:* 1979 to the present
- *HPSG:* 1984 to the present
- *GF*

Categorial Grammars Ajdukiewicz 1935 to the present

...

Chomski's criteria for adequacy of a linguistic theory

- Descriptive level of adequateness Description of linguistic knowledge
- Explanatory level of adequateness
  Explanation of linguistic knowledge is an important target. The ultimate efforts are for comprehensive linguistic theory, which explains the nature of the linguistic knowledge.
- Providing explanation of learnability and what is to know a language, i.e., the ultimately important question

  How is it possible that language is learned?

Note: The Chomskyan focus is on syntax, including morphology and lexicon, while semantics is (somehow) implicit.

Overview of computational semantics

Before and at the beginning of XXth century:

Gottfried Wilhelm von Leibniz (1646-1716)

Friedrich Ludwig Gottlob Frege (1848-1925)

Type Logic Grammars

- *Categorial Grammars:* Ajdukiewicz 1935
- *Type-theoretical Grammars*
- *Montague Grammars:* Montague 1970
- New Type Theories: . . .

Other Approaches to Computational Semantics

- *Situation Semantics*
- *Discourse Representation Theory*
- *Minimal Recursion Semantics* in HPSG
- . . .

Criteria for Adequateness of a Semantic Theory

Semantic Universals of Natural Languages, incl. Human Languages, see [Barwise, Perry, 1983]:
A theory of meaning has to represent adequately the following phenomena:

- Productivity
- Efficiency
  - Proper Efficiency
  - Ambiguity
  - Underspesification
- Perspectival Relativity
- External Significance
- Internal Significance
- Modality (later in the series)
- Intensionality (later in the series)

Semantic notions to be formalized:

- denotations
- interpretations
- Carnap intensions
- referential intensions
- meanings

These notions reflect on (to some extend) the facts that language interpretation depends on:

- context, e.g., on the context of language use
- referential "medium", e.g., references of human users: the speaker's vs. the listener's references

We shall concentrate predominantly on formal and human languages, popular as natural languages (NLs).

Speaker's References

The concept of speaker's reference is a key semantic phenomenon, which requires (at the minimum)

- an utterance (or a discourse) situation: the speaker, the listener, the time, the location, etc.
- a language expression
- an object (real or abstract) referred to and called the referent of the expression
- the speaker's act of the reference in the utterance situation

There are different kinds of reference depending on what language elements are used and what is the logic theory that formalises semantics.

How is it possible to learn to deliver and understand a language?

Syntactic Productivity

A grammar can license infinitely many expressions by using

- a finite dictionary, and
- a finite set of syntactic rules and principles

Chomsky 1957, 1965, 1986

Semantic Productivity: Frege's Principle of Compositionality

The denotation of a compound expression is a function on

- the denotations of its parts, and
- the *way the parts are put together.*

Frege 1892

We will be looking at this principle later on, again-and-again.

How is it possible to learn to deliver and understand a language?

Syntactic Productivity

A grammar can license infinitely many expressions by using

- a finite dictionary, and
- a finite set of syntactic rules and principles

Chomsky 1957, 1965, 1986

Semantic Productivity: Frege's Principle of Compositionality

The denotation of a compound expression is a function on

- the denotations of its parts, and
- the *way the parts are put together.*

Frege 1892

We will be looking at this principle later on, again-and-again.

How is it possible to learn to deliver and understand a language?

Syntactic Productivity

A grammar can license infinitely many expressions by using

- a finite dictionary, and
- a finite set of syntactic rules and principles

Chomsky 1957, 1965, 1986

Semantic Productivity: Frege's Principle of Compositionality

The denotation of a compound expression is a function on

- the denotations of its parts, and
- the *way the parts are put together*.

Frege 1892

We will be looking at this principle later on, again-and-again.

Therefore, a comprehensive computational grammar theory has to provide:

- computational syntax
  - the basic language units (formal/computational lexicon)
  - the syntactic rules that combine constituent structures into more complex ones (formal/computational syntax)
- computational semantics
  - the objects that serve as denotations of the basic language units
  - the operations, by which the denotations of the composite parts are computed from
    - the denotations of their subparts
    - the syntactic operations that have combined them.

How might all this be achieved by a computational, i.e., algorithmic, theory?

Question:

Why do we need parse trees?

Answer:

See what Frege's Principle says.

Recall Some Conclusions about CFGs: w.r. to productivity

- Positive features:
  - finite representation of infinitely many objects by recursion
  - efficient parsing based on CFGs — $O(n^3)$.
- Negative features:
  - syntactic redundancy:
    - spurious parse trees
    - lack of linguistic generalizations
    - redundancy in constraining the selectional co-occurrence restriction (e.g., w.r. to complements and agreement requirements), and by this
    - proliferation into large, non-efficient set of rules
  - semantic deficiency: CFGs are not enough powerful for representing all possible ambiguities, in particular, they are inadequate for quantificational and intentional scoping, etc.
  - descriptively adequate w.r.t. well-formed expressions, but explanatory inadequate

Recall Some Conclusions about CFGs: w.r. to productivity

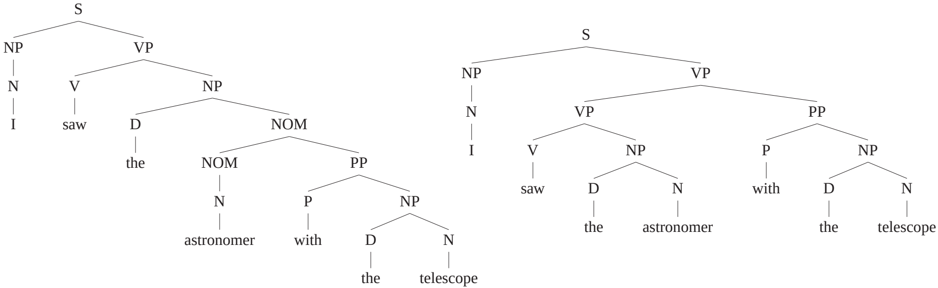
- Positive features:
  - finite representation of infinitely many objects by recursion
  - efficient parsing based on CFGs — $O(n^3)$.
- Negative features:
  - syntactic redundancy:
    - spurious parse trees
    - lack of linguistic generalizations
    - redundancy in constraining the selectional co-occurrence restriction (e.g., w.r. to complements and agreement requirements), and by this
    - proliferation into large, non-efficient set of rules
  - semantic deficiency: CFGs are not enough powerful for representing all possible ambiguities, in particular, they are inadequate for quantificational and intentional scoping, etc.
  - descriptively adequate w.r.t. well-formed expressions, but explanatory inadequate

Proper Efficiency vs. Ambiguity vs. Underspecification

- *Proper Efficiency of NL*
  The same expressions can be re-used in different contexts to transfer different information, e.g., to describe different situations.
  - Indexical and deictic expressions:
    I, you, here, there, ...
- *Ambiguity:*
  - Lexical ambiguity
  - Syntactic ambiguity
  - Combined lexical and syntactic ambiguity
  - Semantic ambiguity
  - Scope ambiguities

Syntactic Ambiguity

In computerized processing, different syntactic parses of the same sentence can represent different denotations.



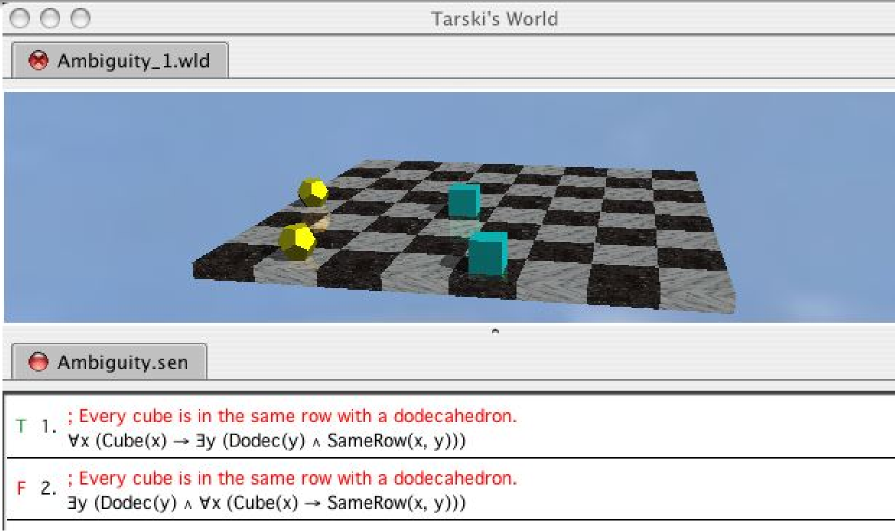
Scope Ambiguity

- There are ambiguities that can not easily be represented syntactically.
- Humans have a unique cognitive ability to resolve ambiguity of sentences by the context of usage.
- The following sentences contain unambiguous words and have similar syntactic structure, but are ambiguous.
  - One of the readings may have higher precedence.

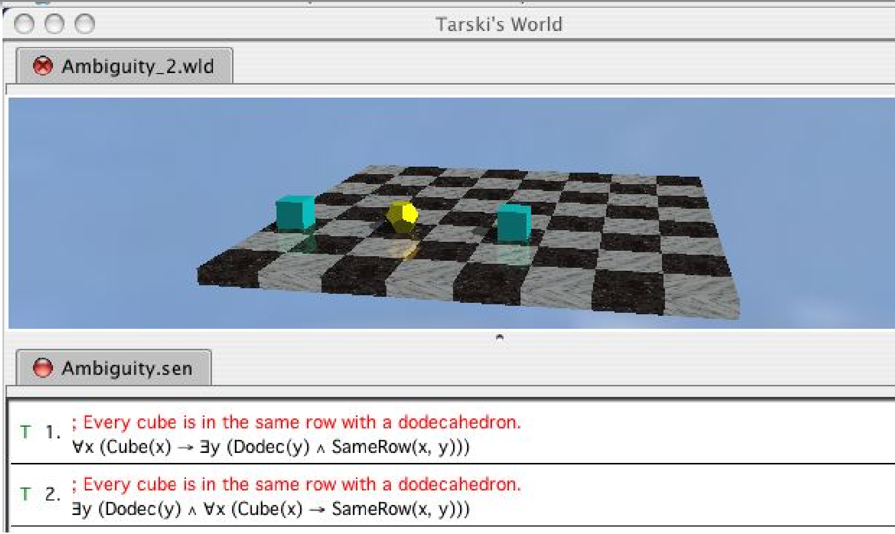
Example

- Every man loves a woman.
- Every student attends a presentation.
- Every cube is in the same row with a dodecahedron.

Ambiguity and Context Dependency: the context resolves the ambiguity



The context resolves the ambiguity. Really?



Ambiguity of NL and Complexity of NLP

- Ambiguity of human languages is a significant part of what makes them hard for computerized processing.

Example

Some representatives of every department in most companies saw a few samples of every product

Hobbs and Shieber (1987)

- There are 42 valid alternative interpretations of this sentence.
- The context of use can resolve ambiguity.
- Do we need to "generate" all possible readings, without any resolving context? The number of the scope readings of a sentence with 5 quantifier NPs is up to 120.
- Formalization of underspecified representations

External vs. Internal Significance of NL

- External Significance Primarily, utterances of NL expressions are used for exchange of information about the real world, by
  - denoting (designating) real world objects with words and expressions
  - describing situations being in a certain way
  - questions, orders, etc., with respect to real world
- Internal Significance
  - language designates mental representations of objects and situations
  - designating and describing abstract objects
  - language as reasoning media

Perspectival Relativity of NL

In an utterance situation, u, a speaker uses a language expression, φ, which is typically a sentence, to describe a situation s as being in a certain way:

$$s \models A$$

where A is a term of a formal language into which φ is translated, e.g.:

$$\varphi \xrightarrow{\text{render}} A$$

- The choice of the language expression φ is dependent on the speaker's perspective in the world, her/his relations to it. From a different perspective, a different expression:

$$\psi \xrightarrow{\text{render}} A$$

- À la Montague intensions (Carnap intensions): functions from possible worlds to objects in them.
  Among the semantic theories covering intensionality are *possible worlds semantics* and *situation semantics*.

Example

- John seeks a unicorn.
- John wishes to catch and eat a fish.
- John believes that The Morning Star is The Evening Star.

- New: The referential intension of a meaningful expression is the algorithm for computing its denotation.

Calculemus!
Let us calculate!

Gottfried Wilhelm von Leibniz

References

Barwise, Jon and John Perry.
Situations and Attitudes.
1983. Cambridge, MA:MIT Press. Republished in 1999 by The David Hume Series of Philosophy and Cognitive Science Reissues.

Hayes, Brian.
Calculemus!
American Scientist. September-October 2008. Volume 96, Number 5, pages 362–366.
http://www.americanscientist.org/issues/pub/calculemus