It's time for a semantic engine!

Ido Dagan
Bar-Ilan University, Israel

Semantic Knowledge is not the goal…

- it’s a primary mean to achieve **semantic inference**!

- Knowledge design should be derived from its usage
  - “Use” comes first, then “Discovery and Organization”

- We should avoid bad experience of developing knowledge for its own sake, where planned usage was left quite vague or as wishful thinking
  - MRDs in 80’s-90’s; inference over WN is hard/questionable

- But how does semantic inference look like?
NLP Applications need morphology

- So… we use an available morphological engine!
  - Namely – a morphological analyzer
    - Input: word (in context)
    - Output: possible morphological analyses (scored/ranked)

- But there are lot’s of morphological phenomena we need to model
  - Yes, but they are all embedded within the morphological analyzer

NLP Applications need syntax

- So… we use an available syntactic engine!
  - Namely – a parser
    - Input: sentence
    - Output: possible parse trees (scored/ranked)

- But there are lot’s of syntactic phenomena we need to model
  - Yes, but they are all embedded within the parser
NLP Applications *need* semantics

- So…. what do we do?
  - Use NER, WordNet, SRL, WSD, statistical similarities, heuristic syntactic matching, detect some negations, etc, etc, etc.
  - Assemble and implement bits and pieces
  - Repetitive application-dependent research – critical mass lacking

- Observations:
  - Logic/formal semantics not widely adopted
  - Common paradigm – miscellaneous interpretations
    - “Semantic annotations/analysis/parsing”

- Can we have a *generic* semantic inference engine?

Engine Desiderata

A computational engine for a target level of language processing should provide:

1) *Generic module for applications*
   - A common underlying task, unified interface (API)
2) *Unified paradigm for addressing different sub-phenomena*
3) *Unified knowledge representation(s)*
Talk Outline

1) Textual Entailment and why it may be a good framework for semantic inference
   • As presented in previous talks

2) A proposal for a concrete engine – API

3) Research challenges towards such engine

4) Requirements for semantic knowledge

⇒ New step of this talk:
   aim at concrete engine, move beyond framework

Textual Entailment:
Task and Framework
Natural Language and Meaning

Why is it difficult?

Variability of Semantic Expression

The Dow Jones Industrial Average closed up 255
Dow ends up
Dow climbs 255
Dow gains 255 points
Stock market hits a record high

Model variability as relations between text expressions:
- Equivalence: text1 ⇔ text2 (paraphrasing)
- Entailment: text1 ⇒ text2 – the general case
Typical Application Inference

**Question**
Who acquired Overture?

**Expected answer form**
$X$ acquired Overture

Application inferences can be reduced to entailment

- **IE:** $X$ acquire $Y$
- “Semantic” IR: Overture’s acquisition
- Summarization (multi-document): identify redundant info
- MT: paraphrasing, evaluation
- Educational applications: student answer vs. reference

Applied Textual Entailment

- Directional relation between two text fragments: Text ($t$) and Hypothesis ($h$):

  $t$ entails $h$ ($t \Rightarrow h$) if humans reading $t$ will infer that $h$ is most likely true

- Operational (applied) definition:
  - Human gold standard
    - Entailment judgment matches applications judgments
  - Assuming common background knowledge
    - Language & world knowledge
Evaluation: PASCAL RTE Challenges

<table>
<thead>
<tr>
<th>TEXT</th>
<th>HYPOTHESIS</th>
<th>TASK</th>
<th>ENTAILMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Regan attended a ceremony in Washington to commemorate the landings in Normandy.</td>
<td>Washington is located in Normandy.</td>
<td>IE</td>
<td>False</td>
</tr>
<tr>
<td>2 Google files for its long awaited IPO.</td>
<td>Google goes public.</td>
<td>IR</td>
<td>True</td>
</tr>
<tr>
<td>3 ...: a shootout at the Guadalajara airport in May, 1993, that killed Cardinal Juan Jesus Posadas Ocampo and six others.</td>
<td>Cardinal Juan Jesus Posadas Ocampo died in 1993.</td>
<td>QA</td>
<td>True</td>
</tr>
<tr>
<td>4 The SPD got just 21.5% of the vote in the European Parliament elections, while the conservative opposition parties polled 44.5%.</td>
<td>The SPD is defeated by the opposition parties.</td>
<td>IE</td>
<td>True</td>
</tr>
</tbody>
</table>

Biased datasets possible for individual phenomena and applications
- temporal, spatial, quantification, compounds, …

RTE Participation and Impact

- Very successful challenges, world wide:
  - RTE-1 – 17 groups
  - RTE-2 – 23 groups
    - ~150 downloads
  - RTE-3 25 groups
- RTE-4 (2008) – under NIST
  - New TAC conference (with QA and summarization)
- High interest in research community
  - Papers, sessions and areas, PhD’s, funding proposals
  - Special issue at of NLE Journal, ACL-07 tutorial, AVE@CLEF, …

- There must be something into it…
Initial use of RTE systems in applications

- QA
  - Harabagiu & Hickl, ACL-06
  - Answer Validation Exercise at CLEF
- Relation extraction
  - Romano et al., EACL-06
- Educational applications
  - Nielsen et al., ACL-08 workshop
- Ongoing projects

Textual Entailment vs. Classical Semantic Approach
Classical Approach: Semantics as an Interpretation Task

- Stipulated Meaning Representation (by scholar)

- Variability

- Language (by nature)

- Logical forms, word senses, semantic roles, named entity types, … - scattered tasks
- Feasible/suitable framework for applied semantics?

Textual Entailment = Text Mapping

- Assumed Meaning (by humans)

- Variability

- Language (by nature)
General Case – Inference

- Entailment mapping is the actual applied goal
  - Adopt it as a unified task for semantic processing
  - Theory neutral !!!

- Interpretation becomes a possible mean
  - Direct inference at language level may be attempted and formalized

Towards a Concrete Semantic Engine
Entailment-based Inference Engine

- Base a generic inference engine on the entailment task
  - Application inferences reducible to entailment API
- Different than traditional approach of semantic analysis/annotation
  - E.g. semantic parser/SRL
  - Assuming that applications would make their own inferences, facilitated by these annotations

→ Provide the eventual inference by the generic engine
  - Avoid forcing semantic annotations on the application (vs. morphology & parsing!)

- Following slides specify initial API and its use by applications

Basic Task - Recognize Entailments

- RTE challenge mode
- Given \((t,h)\) pair:
  - Is \(h\)'s truth entailed/contradicted/unknown based on \(t\) ?

- Validation mode for applications
- Target meaning reduced to \(h\):
  - QA – candidate answer entailed by passage
  - IE – candidate extraction entailed by passage
  - MT evaluation – equivalence of translation and reference
  - Tutoring – student answer entails reference
  - Summarization – avoid entailments within summary
Typed Variables in Hypothesis

- Address “slot filling” (e.g. QA, IE)

- Allow variables in $h$ (template hypothesis)
  - $h$: $X$ elect $Y$
  - $t$: a candidate text for an election event

- Entailment engines can be naturally extended to instantiate variables

- **Textual typing** for variables
  - Types are unrestricted textual expressions
    - For both names and terms (vs. pre-defined NER types)
  - Election example: <$X$: executive board, $Y$: company official>
  - QA: “which” questions (river, president, treatment, …)

Meaning Disambiguation via Contextual Preferences

- Specify/disambiguate the meaning of the target $h$
  using *textual* context information

- Context types:
  - Textual types for variables (as above)
  - Disambiguating context terms
    - IE: $X$ attack $Y$ (military, war, terrorism)
    - Session/user context in QA&IR, doc context in summarization

- Avoid stipulated annotations to specify meaning –
  stick to textual representations
  - *Contextual Preferences* – Szpektor et al., ACL-08
Operation modes:
Recognition, Search, Generation

- Recognize entailment given $t/h$ pair
  - RTE mode – validation
- Given $h$ and corpus/doc, find all entailing texts
  - QA, IR, IE against corpus/doc
- Given text, generate all entailed statements
  - Paraphrase generation for MT
- Identify entailments of parts of $h$
  - Summarization, partial match

API Summary

- Recognizing entailment for a $(t,h)$ pair
- Variables with textual types in hypothesis
- Contextual preferences (textual)
- Support all possible operation modes
Research Challenges

- **Knowledge acquisition**
  - Learning (corpora and web), extraction from resources, manual

- Inference
  - Principled models (logic & probabilistic inspired)
  - Representation, efficient search, scoring, approximate match

- Model specific semantic phenomena
  - Derivations, compounds relations, typing, modals, polarity...

- Utilizing generic entailment technology for applications

- Other NLP infrastructure
  - Parsing
  - Co-reference

*Teaser: current steps at BIU – illustrate feasibility (poster)*

---

Desired Characteristics of Semantic Knowledge

*Joint community resource?*

- Human genome analogy
- Contributing output of automatic & manual acquisition
  - Web extraction with search engines
  - Public manual KE environments, Mechanical Turk
- Very preliminary: ACL-Wiki RTE Resource Pool

*Requirements:*

- Unifying representations and interfaces for diverse knowledge
  - Research needed to formulate them
    - E.g. entailment rules
  - Allow modules encapsulating knowledge (on the fly via API)
- Theory neutral
  - Minimize stipulated representations, and need for conversions
Conclusions

- It’s time to think of generic semantic engines
- Should provide needed inference for applications
- My proposal:
  - Base engine on entailment rather than annotation
  - Initial API
  - Highlights requirements of semantic knowledge

- Other suggestions are welcome…
- How long did it take for (somewhat) useful parsers?

Thank You!