

The SUBTLE NLP Pipeline (and an integrated system preview)

Constantine Lignos

SUBTLE MURI Year 4 Review

10/21/2011

Goal: Natural language to a logical controller

Commander's instructions

```
graph TD; A[Commander's instructions] --> B[Parsed sentences]; B --> C[Semantic representation]; C --> D[MetaPAR/LTL]; D --> E[Automaton];
```

Parsed sentences

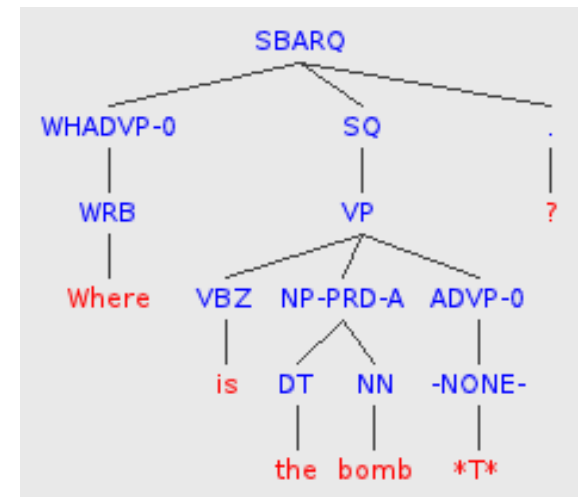
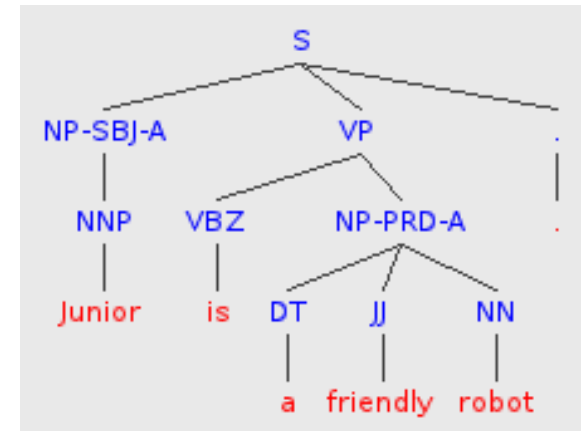
Semantic representation

MetaPAR/LTL

Automaton

How do we get there?

- Pipeline of systems:
 - Tagging: Assign a part of speech tag to each word
 - Maximum Entropy Tagger (Ratnaparkhi, 1997)
 - Parsing: Assign a syntactic structure to the sentence
 - Head-driven lexicalized parsing (Collins 1999; Bikel, 2004)
 - Additional function tags (Kulick, 2006)
 - Null element restoration: Reconstruct movement of words in parse (Gabbard, 2010)



Limitations of the pipeline

- Models are still built on Wall Street Journal text
 - Poor handling of imperatives
 - Reliance on punctuation
- Feed-forward system means that parse/tagging errors are fatal and cannot be recovered from
- Demonstration of system
 - Keep in mind parsing components span 15+ years of work and 3 PhD Theses, while semantics work is very young

Brief demonstration

Moving forward

- User interface for exploring parsing allows us to see the limitations of the current system and identify weak points
- Retraining models on more appropriate data is the next major step

Broadening the audience

- We want to make these tools publicly available!
- Web demo to show how the components perform
- Open source code releases planned:
 - MXPOST: Release-ready
 - Modified Bikel Parser: Currently reviewing modifications to verify they're release-ready
 - Gabbard Null Element Restorer: Some work needed before release

A Preview of the Integrated System

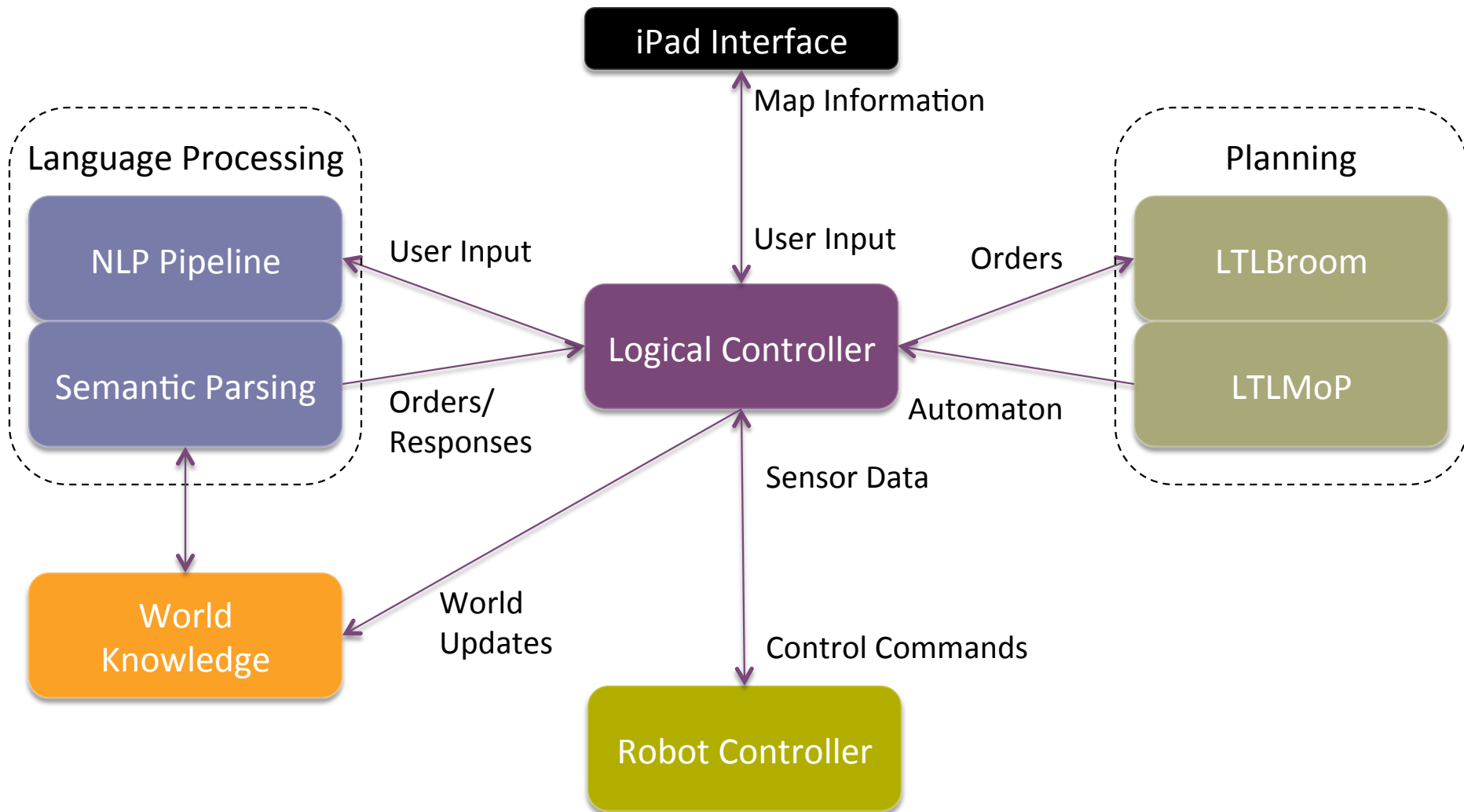
Rewind: Goals of last year's integrated system

- Give a basic end-to-end demo of the current software components in the project
- Demonstrate the following:
 - Specifying a plan using natural language
 - Naturally answering questions about the robot's state
 - Visualization of the controller being executed and interacting with the simulated robot
- Provide a software structure for more detailed development

Steps from last year's systems

- Two silos:
 - Natural language → LTL → Automaton → Simulated robot (Penn)
 - Structured language → LTL → Automaton → Real robot (Lowell)
- Clearly, it was time to connect the two together: Natural language → LTL → Automaton → Real robot
 - Easy, right? Not at all, but a lot was learned.
 - Key issue: we had not found the right way to represent the “real” world in the logical world in a consistent fashion
- We found the assets on each side could not scale the way we'd like: time to design a new system!

Integrated system overview



Natural language major improvements

Focus: Scalable systems ready for integration with more sophisticated pragmatics, able to handle a known map and frontier-based exploration.

- Ability to execute complex plans, previously: fixed search plan, only giving standing orders
- Integration of more LTLMoP assets
- Replacing keyword-spotting semantics with more sophisticated system
- World knowledge to serve as basis for pragmatic inference

Integrating with multiple environments

Goal: *One code base for web interface and multiple robot systems*

Results:

- Deep integration with Lowell system (demo later)
- Integration in progress with ARL Symbolic and Subsymbolic Robotic Intelligence Control System (SS-RICS)
 - ARL's .NET-based system proved particularly challenging
 - All SUBTLE NLP Pipeline components running in SS-RICS
 - Semantics running in SS-RICS
 - Limited natural language commands functional
 - World knowledge still needs to be integrated

SUBTLE NLP in SS-RICS

The screenshot displays the SS-RICS NLP interface within a Windows 7 Parallels Desktop environment. The main window, titled "Natural Language Input", shows the input sentence "Go to the hallway." and the resulting parse tree. The parse tree is a hierarchical structure of nodes representing syntactic categories. The root node is "S", which branches into "NP-SBJ-A", "VP", and an empty node. "NP-SBJ-A" leads to "-NONE-". "VP" branches into "VB" (which leads to "Go") and "PP-CLR". "PP-CLR" branches into "TO" (which leads to "to") and "NP-A". "NP-A" branches into "DT" (which leads to "the") and "NN" (which leads to "hallway").

Other components of the interface include:

- Map:** A floor plan of a building with a blue arrow pointing to a specific location.
- Edit Goals:** A menu with "File", "Edit", and "Debug" options, and a list of goals: "gDidNotUnderstand", "gDontKnowHowToX", "gGotoX", "gSayGoingNowhere", and "gSayGoingToX".
- Semantics Output:** Shows "Commands: go: Hallway" and "Frames: ('Location': Tree('NP-A', [Tree('DT', [the]), Tree('NN', [hallway])]), 'Agent': Tree('NP-SBJ-A', [Tree('-NONE-', [])], 'VERB': Tree('VB', [Go]), 'PREP': Tree('TO', [to])])".
- Commands generated:** Shows "[go', ['Hallway']]"
- Taskbar:** Shows various application icons including "SubSimNL2 ...", "Rick (SS-RICS)", "Natural Lan...", "Edit Goals", and "MobileSim".
- System Tray:** Shows the time "7:55 AM" and date "10/21/2011".

Summary

- Junior's NLP system is growing up
 - Mature assets close to public release
 - Semantics growing quickly in scope and capabilities
 - User interface to identify issues and improve system
- Significant integration progress year over year
 - Language, logical, and robotic systems tightly connected
 - Integrating richer pragmatics is the next frontier
- Promising but still nascent work with ARL cognitive robotics group