The Pipeline: an Introduction

Natural Language Text > Extract Events and Semantic Features > Infer temporal ordering between events

Infer logical relationship between events > Construct a planning problem > Generate natural language text from solution to planning problem
Current Progress

Natural Language Text → Extract Events and Semantic Features → Infer temporal ordering between events

Infer logical relationship between events → Construct a planning problem → Generate natural language text from solution to planning problem
Current Work: Event Extraction

- The basic idea stems from James’ work. Events can be found in natural language English sentences expressed as a triple:
  - `<Entity Action Entity>`
  - This stems from the structure of sentences in English which often takes the form
  - `<Subject Verb Object>`
StanfordNLP Parser – Doing the heavy lifting

- Generates a syntactic parse tree
- Creates a dictionary for each token (such as POS, lemma)
- Creates a list of dependencies which make up the parse tree
- Our job is to extract a triple that can represent a summarized meaning (to humans and machines) represented by the sentence and that can be turned into a time-ordered log of events and then a logical model later
Event Extraction: a deep dive into natural language

• Work consisted of applying James’ triple model for events into code

• Right now: a rudimentary event extraction system has been implemented. It can find events from simple sentences.
What’s Next?
Test and debug advanced function that deal with complex traversing of the parse tree
Aim for this Pipeline Section

- To make this code reusable and documented, explicitly for future extensions
  - Jupyter Notebook is an excellent tool for this
  - There is room for further additions from a linguistic point-of-view
Next Stage

- Temporal Ordering Stage, use machine learning mechanisms
- Creating a logical model
Questions