Introduction to Natural Language Understanding

(incorporating structured and opportunistic approaches)

Sections 6.3, 7.1,7.2, Chapter 14 (omitting 14.3,14.4)

Natural Language Understanding

- One of the earliest efforts in AI was to develop systems that could understand natural language; e.g. the Eliza system
- Well-respected researchers stated that machines would be able to understand natural language within 15 years
- These efforts (obviously!) did not meet with success

Natural Language Understanding

- The difficulties of dealing with context and breadth were grossly underestimated
- Recently, NLU systems have been more successful as they incorporate more domain knowledge
- Natural language text generation is a simpler task
- e.g. generate an explanation in English of the rules that were used to solve a problem in an expert system

NLU vs. Speech Understanding

- We tend to think of speech and language as being very similar, just because speaking and listening are natural things to us
- But actually understanding speech as opposed to reading text is a much more significant problem
- Really we're adding perception to the reasoning problems already inherent in NLU
- We've seen how difficult perception can be!

Speech Understanding

- We've also seen that context plays a huge rule in perception (e.g. knowing we're in a hallway makes a robot look for a second wall where it's seen one already)
- Similarly, much of understanding speech in a noisy environment is about using context to fill in gaps where what is heard is uncertain
- We all have the experience of saying "what?" to somebody and then before they even answer, realize what they were talking about

Natural Language Understanding

- The goal of an NLU system is to understand a collection of words
- The words are presented in written form (i.e. machine readable: each word is a separate unit, punctuation is included)
- The system produces a symbolic description of the meaning of the words
- Understanding is association with known (symbolic) concepts – that's usually what we mean when WE say we understand something

Basic Model of NLU

- Natural language understanding involves several phases, looking from the lowest level up:
- Morphological analysis:
  - determine the type of each word (e.g. noun, verb, etc.)
- Syntactic analysis:
  - determine the structure of the words: phrases, sentences

Basic Model of NLU

- Semantic analysis:
  - determine the meaning of the individual phrases
- Pragmatic analysis:
  - determine the meaning of the phrases together (the utterance or passage as a whole)

- We'll look at a simple breakdown of each step (though each can itself get very complicated!)

Morphological Analysis

- Morphological analysis involves determining the type of each word
- The processing involves determining the primitive units (morphemes) of each word
- For example, many words can be broken into
  - [prefix] root [suffix]
- incoming $\rightarrow$ in come ing
- Not all words break up in this manner, and the root can be disguised sometimes
- Worse in some languages than others

Morphological Analysis

- You can see why we took basic search stuff before looking at applications, because you should see how it can be applied here (e.g. looking at possibilities for breakdown)
- Strong procedural components here as well: looking roots up in a dictionary, templates for many words that can save us search
- May have several hypotheses (uncertainty here too!) if the use of the word is ambiguous
- Output is a breakdown into morphemes, label of word type, and the looked-up meaning of each word

Syntactic Analysis

- Syntactic analysis involves determining the structure of the words in each sentence
- uses surface reasoning; does not determine the meaning of the words
- The goal of syntactic analysis is to produce a symbol structure that indicates the word groupings
- Note that often the morphology won't be accurate until this is – some words can be used as nouns or verbs, for example

Syntactic Analysis

- Syntactic analysis involves parsing the words according to a description of the valid combinations of word types
- Pure syntactic analysis is performed without any domain knowledge -- this may result in phrases that are ambiguous and cannot be parsed completely
- Part of the power of our natural language is its flexibility – but this flexibility is just what makes NLU difficult

Syntactic Analysis

- One common method used to parse sentences is using a grammar, as we would artificial languages such as programming languages
- The grammar consists of rules that define the valid combinations of words
- The grammar defines the parts of each sentence and the valid terminal symbols that may occur

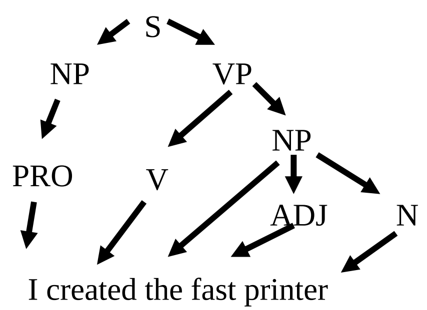
Syntactic Analysis

- A simple grammar:

```
S → NP VP
NP → the ADJ N
NP → PRO
VP → V
VP → V NP
N → file | printer
ADJ → short | long | fast
PRO → I
V → printed | created |
```


Syntactic Analysis

- The result of parsing a set of sentences is a parse tree of the components of each sentence



- Note this is a trivial grammar!

Syntactic Analysis

- Parsing can be carried out in a top-down manner (backward search)
- begin with the S symbol in the grammar
- Parsing may also be carried out in a bottom-up manner (forward search)
- begin with the words and match them to the terminal strings in the grammar; work towards the S

Syntactic Analysis

- At this point we have an uncertain morphology and an uncertain parse tree (possibly many candidates)
- And we still haven't gotten into the meaning of much of this at all, we're just putting the pieces together
- Really, reducing the enormous uncertainty of meaning by structuring the problem and what we know about it before attempting to analyze meaning

Semantic Analysis

- Semantic analysis determines the meaning of the words in each of the phrases: deep reasoning
- Semantic processing creates a symbolic description of the meaning of the phrases
- How do we do this? If understanding means the association between ideas or concepts, we need to give our system concepts to associate
- e.g. "The cat came back" vs. "The boomerang came back" – it needs to know the difference between these two objects to understand the context

Describing Concepts

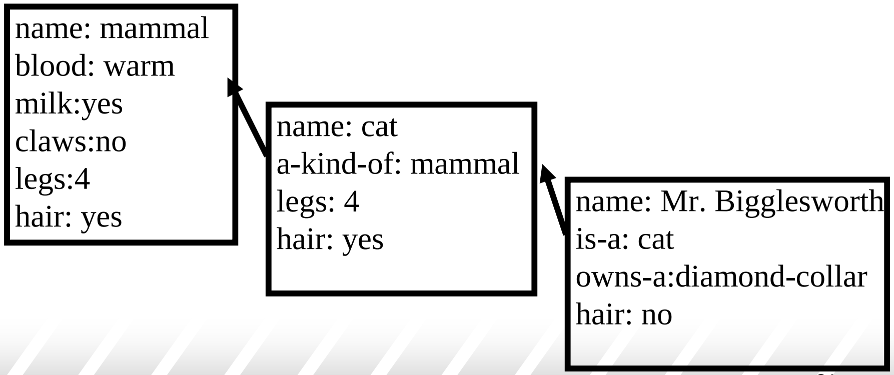
- If I want to describe some object or process, this obviously is made up of many different components and connections to others
- Cats: colour, size, attitude, the fact they're alive, fragile, bite, etc…
- You can envision describing these with rules and facts, but this becomes tedious
- Too many individual items
- No organization – our cat facts blend with boomerangs and 10,000,000 other things

Describing Concepts

- Because of this we use Structured Representations for deep reasoning (and also for convenience in shallow reasoning)
- Structured representations lay a structuring onto the basic facts and rules
- This structuring provides an organization for the use and management of this knowledge
- These basic ideas were experimented with in AI 40 years ago, and led to today's object oriented approaches that you're familiar with
- they tend to be much more dynamic, however

Frames

- Frames are the most basic type of structured approach in AI – they provide inheritance as you'd expect



Frames

- Search for a value starts bottom up, allowing specific instances to override general defaults
  - You see the ease of doing default reasoning with these!
- Search goes from instances to higher level objects through specially named slot connections: is-a (instance relationship), a-kind-of (subclass relationship)
- Also possible to define our own such links for broad multiple inheritance: e.g. containment, possession, causality relationships)

Frames

- All this is DYNAMIC – we can change any slot at any level at any time, and the behaviour changes
- e.g. by changing the *a-kind-of* slot, we can shift inheritance completely – not normally possible in (static) object-oriented languages
  - but necessary for flexibility of representation here
  - so somebody talking about their mercury could let you shift your concept from a kind of car to a kind of chemical…
- Procedural attachments a la OO also possible

Structured Approaches and NLU

- Frames are most normally see in expert systems
- In NLU, an analogous structured approach called Conceptual Dependency Theory is often employed
  - specialized for the kind of thing we need to represent in NLU
- Conceptual dependency theory associates each phrase with the primitive acts that the phrase describes
- Primitive acts are structured in a fashion similar to the way that frames structure objects

Conceptual Dependency Theory

- There are numerous primitive act categories:
- physical acts: move, propel, ingest, expel, grasp
- state change acts: change the location of a physical object; change in an abstract relationship (e.g. sell a house)
- communication acts: speak, attend (receive information)
- mental acts: remember, recall, think

Conceptual Dependency Theory

- These primitive acts are combined into very explicit conceptual dependency structures that indicate the purpose of each word in the phrase
- Defining the primitive actions in conceptual dependency theory is much more complex than defining a syntax grammar
- The meaning of each type of phrase must be defined completely (at least so far as we want the system to understand it!)

Conceptual Dependency Theory

- Most acts refer to an agent, an object, a source, and a destination:

"John threw the ball into the yard."

- This information can be represented in a frame structure

Act:	Move Object
Agent:	John
Object:	ball
Source:	John's hand
Destination:	the yard
Time:	past

Semantic Analysis

- Semantic analysis involves assigning a meaning to the individual phrases (or events)
- "Fred wanted to play baseball but it rained."
- "Fred wanted to play racquetball but it rained."
- Semantic analysis would generate the same semantic structures for these sentences even though a different meaning is intended:
  - Fred desired some activity, but this did not occur – the fault of this was because of the weather

Semantic Analysis

- this can sometimes be much more subtle than the previous example:
  - "John wanted to golf but it snowed" vs
  - "John wanted to go bowling but it snowed"
  - The latter is an indoor game that weather should not interrupt – however, there is the possibility that the required travel to the bowling alley is impossible

Pragmatic Analysis

- Pragmatic analysis determines the meaning of the phrases as a whole (the gestalt)
- The system requires a detailed domain model in order to be able to infer information that is not stated explicitly
- This goes far beyond what we needed in semantic analysis – we need explicit domain knowledge to fill in information that is NOT included in the utterance
- The model must contain knowledge about actors, situations, props, expectations, etc

Pragmatic Analysis

- Consider the sentence:
- "Mary drove to the restaurant, had lobster, and then left for the movie."
- What time of day is it?
- Did Mary pay for the lobster?
- Was the lobster cooked?
- Did Mary eat ONLY lobster?
- Did Mary leave her car?
- We infer a tremendous amount of information from this sentence because of we are familiar with restaurants

Scripts

- We need another specialized representation for this
- Scripts are a structured form of knowledge that attempt to describe the goings-on in a domain – what normally happens, what to expect
- All the world really is a stage in this sense
- Like a theatre script, these involve describing actors, props, settings, and sequences of expected events
- To understand the example of Mary, we need to understand what happens in specific types of restaurants

Scripts

- As we said Scripts are structured
- This means we link scripts together
- Going to a restaurant involves getting there (temporal connections)
- Each script contains lower-level acts
- Eating is a script: not normally a thrilling sequence of events, but certainly a set of expectations placed on behaviour
- Eventually these break down into the primitive acts of conceptual dependency theory

Pragmatic Analysis

- Involves applying this domain knowledge to the structuring we've already produced to provide an understanding of the phrases
- We invoke one or more scripts that allow us to understand elements that aren't specifically explained in the dialogue
  - also uncertainty management involved here; e.g. type of restaurant
- …and also to further resolve ambiguities in interpretation!

Pragmatic Analysis

- We can then ask questions (like whether Mary paid for her lobster), and link an overall dialogue together
- This is an example of the kind of specialized knowledge that is needed to apply AI techniques to complex tasks
- You can see the general techniques we started out with would fail miserably!!!
- Scripts are also one approach to representing common-sense knowledge for a specific purpose
  - might not be so useful for other purposes!

Pragmatic Analysis

- Obviously the most difficult (which is why we try to pare down the number of possibilities to this point)
- But also the phase where the most interesting elements happen – where we answer the most difficult questions
- e.g. deciding which of many possible interpretations is valid for complex phrases; for example: "*Sam saw Diane in the park with a telescope*"
- Is Diane an astronomer or is Sam creepy?
- A significant part of complex language use in humour, poetry, songs, etc. involves dealing with this level, and would likely be completely incomprehensible without it

NLU Summary

- There is still a significant amount of knowledge that is needed before an NLU system is able to function at or near the human level
- discourse knowledge: knowledge of how people converse
- belief knowledge: knowledge about the (differing) beliefs of people
- **commonsense knowledge about the world and the way that people and objects in the world function – this is a huge undertaking!**
  - see Cyc!

NLU Summary

- NLU involves very complex processing that can not easily be subdivided into neat packages (syntax analysis etc.) that are independent of each other

Interaction Between Levels

- I've presented this as separate levels – but really they're heavily interactive
- As we said, syntactic knowledge alone leaves us with many candidate interpretations
- You can look at resolving those (finding the correct one) using later stages
- This is why speech understanding is so much harder: think of the number of candidates we get when we go to a lower level (recognizing parts of syllables!)

Opportunistic Approaches

- It turns out working completely bottom up isn't a useful way of doing things – too much uncertainty, too many candidate interpretations, especially at lower levels
  - e.g. whether a word is a noun or a verb
- There are times where we appear to work top down ourselves (e.g. when we miss a word, we fill it in with likely candidates given the context of the dialog)
- But completely top down would be silly too!

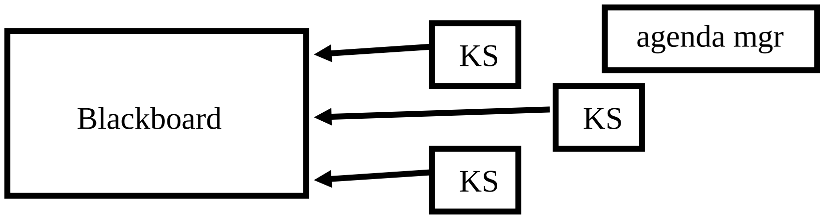
Opportunistic Approaches

- Really what we want to do is work in both directions, and at any point choose what makes the most sense to do – e.g. hypothesize a couple of word usages, then see if they work in a grammar, and so on
- We want to recognize opportunities to make large solution steps (e.g. recognize when a certain word would fit in context even if we haven't recognized it
  – we do this all the time: opportunistic search as opposed to purely forward/backward

Opportunistic Approaches

- Rely on an agenda of some sort: we do the most important thing at any point and modify the agenda as we go to reflect new opportunities
- A blackboard approach is the most commonly used of the opportunistic approaches
- Metaphor based on the idea of a cluster of experts working around a blackboard, with one piece of chalk
- Everybody tries to contribute when we can, the agenda decides on which expert gets to go

The Blackboard Model



- Our experts are called knowledge sources: each has a set of preconditions indicating when it's useful
  – A grammar KS might say it's useful when we have a morphology of a particular word that is uncertain between two word types (e.g. party as a noun and a verb)
- Hypotheses, facts are kept on a global blackboard accessible (and alterable) by all

Knowledge Sources

- The agenda manager manages a list of potential knowledge source activations
  – When a KS's preconditions are met, it can place itself on this agenda
  – The agenda is ranked by the agenda manager (this may itself be a knowledge source – knowledge about which experts are useful when!)
    - agenda is reranked after each KS run
  – When it's at the top of the agenda, the KS can use it's specialized reasoning approach to work and alter the blackboard, potentially stimulating other experts

Similar to?

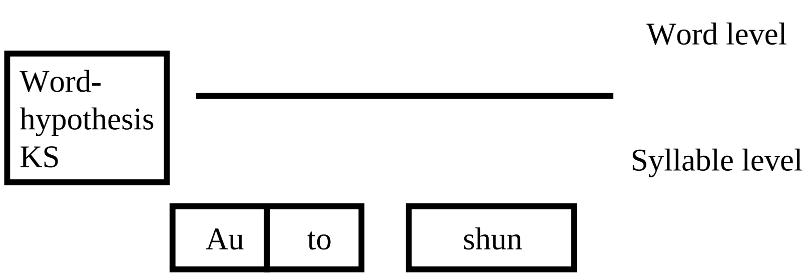
- Rule-Based systems
  - Condition-action components in KS's
  - BB -> working memory
  - Agenda manager: conlict resolution strategy
- Difference?
  - Granularity
  - Here we're talking about complete systems specialized for various purposes
  - Think of interacting expert systems!
  - This is really a primitive example of a Multi-Agent approach to problem solving

Multi-Level Blackboards

- Most systems partition the blackboard into multiple levels
  - Allows knowledge sources to effectively move data from one level to another
  - Many problems are naturally partitioned in levels: NLU!
- e.g. Hearsay-II - a speech understanding system
  - Blackboard contains hypotheses regarding the meaning of an uttered phrase
  - Divided into many levels: candidate syllables, words, partial phrases
  - support from lower levels can drive support for higher-level hypotheses

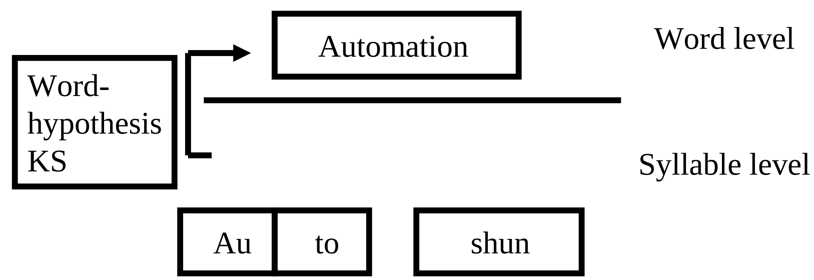
Hearsay example

- Example - somewhat contrived



Hearsay example

- Example - somewhat contrived



  - islands of certainty
  - As we have said, we observe processes like this in NLU (and in signal processing in general!) all the time