Natural Language Processing (NLP)

Our goal is to study how we might build an agent that can understand natural language utterances. The agent must translate from, e.g., English to its internal knowledge representation language (logic). Before we start to look at this properly, let's briefly draw a distinction between 'shallow' and 'deep' NLP.

1 'Shallow' and 'Deep' NLP

In recent years, a particular approach to NLP has been gaining favour. In the short-term, this approach may be very useful for building practical NLP systems for simple tasks, where a proper understanding of the language is not required. I call this a 'shallow' approach to NLP. Here are two examples of this 'shallow' approach.

Suppose you have a large number of documents and you wish to offer a search engine to help people to select relevant documents. Obviously, this is a task much looked at by Information Retrieval (IR) researchers. For the most part, IR systems retrieve documents based on matching keywords from the user's query with words in the document. But this can be too simple and may be prone to retrieving too many unwanted documents. One way of trying to improve the retrieval accuracy of the system is to make use of some 'shallow' NLP.

Suppose you had a way of taking some text and, for each word in the text, determining whether it is a noun, a verb, a preposition, etc. You might refer to this as *tagging* the text. How might it be done? Well, one way is to get some human beings to manually tag a very large number of documents, and then run a machine learning algorithm over these manually tagged documents. The learning algorithm might find, for example, that the word *"walk"* is usually a noun if preceded by the word *"the"* but it is usually a verb if preceded by the word *"you"*. The knowledge that this machine learning algorithm acquires can then be used to tag unseen texts.

Now, armed with the ability to automatically tag documents, you can tag all the documents in your library and tag the user's query. Then your retrieval algorithm can look for matches between words and their tags. Maybe this will improve the quality of the retrieval. (And maybe it won't.)

Here's a second example. Suppose you need to translate documents from one language to another. You may wish for a machine to help you with this task. One ('shallow') approach is word-for-word translation. This is at the heart of the approach used by the translation facility offered by several Internet search engines. But there is a slightly more sophisticated (but still 'shallow') approach. Again it involves machine learning.

Suppose you have access to large quantities of text in one language and, alongside it, parallel text in another language. In Canada, for example, the proceedings of the parliament are published in both English and French. From these parallel texts, a machine learning algorithm can discover correspondances. For example, it might learn that the word *"course"* is translated by *"classe"* when nearby words include *"student"* and *"university"*, but it is translated by the word *"plat"* when nearby words include *"menu"* and *"dinner"*. These correspondances can be used to translate unseen texts.

In neither of these examples (the IR or the machine translation) is any attempt being made to recover the meaning of the texts. And this is why I call them 'shallow' approaches. They may be adequate as practical systems. But it is obvious that if we could make 'deeper' approaches more practical, then they would be more successful: search engines that could understand the meaning of the documents and the intention behind the user's query or machine translation systems that could understand the meaning of text would surely perform better. Such systems are not really available now. But in these lectures we'll look (all too briefly) at the approach being taken.

In the remaining sections of this lecture, we give an overview of the different types of linguistic knowledge that may need to be represented and processed in 'deep' NLP systems. We also describe the processing modules responsible for manipulating each type of knowledge in a system that understands natural language, and, because of their impact on the difficulty of NLP, the ambiguities that may be discovered at each processing phase.

2 Below the level of the word

Suppose the sensory devices of our agent receive a (continuous) speech waveform, which it digitises. Then it has the task of breaking this signal up in such a way as to determine what words there are in that signal.

Ambiguities can be found during this segmentation process. Both sentence (1) and sentence (2), for example, could result from different segmentations of the same signal:

(1) *"It's a grade A."*

(2) *"It's a grey day."*

We're not going to look any further at this part of the process. We're going to simplify and assume that our agent is given a sequence of words.

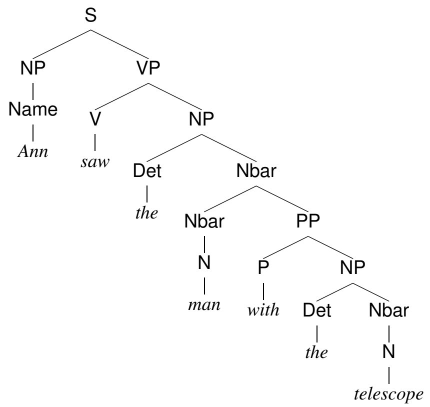
But words themselves have internal structure. They are often made up of smaller meaningful units. For example, the word *"unwillingly"* is made up of *"un"*, *"ing"* and *"ly"* as well as the root of the word, *"will"*. Dictionaries (or, as they are more often called in NLP, *lexicons*) often contain entries only for word roots. Humans (or machines) who want to look up words may have to know how to extract the root from the word before searching the lexicon. Again, this is something we'll ignore in our brief treatment.

3 Syntax

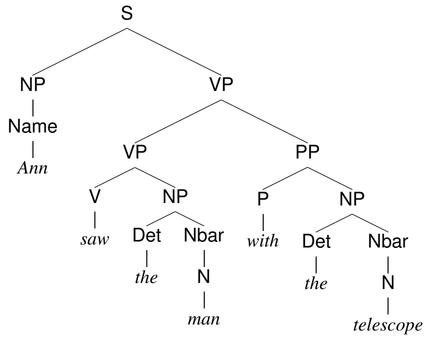
Words can be put together to form bigger units, known as phrases and sentences. The rules that govern the structural aspects of this are referred to as the *syntax* of the language. Syntax is exclusively concerned with *structure* and not with *meaning*. Syntax tells us, for example, that *"the dog barked loudly"* is grammatical, i.e. is structurally well-formed, but that *"dog the barked loudly"* is not.

Syntax also tells us how grammatical strings are themselves made up of other grammatical strings. For example, the sentence *"the dog barked loudly"* is made up of the two structurally well-formed phrases *"the dog"* and *"barked loudly"*. In other words, syntax assigns an internal structure to sentences. We call this the *phrase-structure*. And syntax tells us the grammatical role that words and phrases play within a particular phrase-structure by assigning *syntactic categories*, such a noun (N), verb (V), noun phrase (NP), prepositional phrase (PP), to words and phrases.

The phrase-structure of a grammatical expression can be shown as a *parse tree*, e.g.:



There is the potential for ambiguity here. A sentence might have more than one feasible structural analysis, which will then give rise to multiple meanings. The sentence above, for example, has an alternative phrase-structure:



Syntactic knowledge is captured in a *grammar* and a *lexicon*. *Context-free phrase-structure grammars* (CF-PSGs) are one simple kind of grammar that we will use here for illustrative purposes. Here is a CF-PSG and its lexicon for a fragment of the English language.

Grammar	Lexicon		
S → NP VP	*Ann* : Name	*I* : Pro	*they* : Pro
NP → Name	*Ben* : Name	*me* : Pro	*them* : Pro
NP → Pro	*man* : N	*we* : Pro	*with* : P
NP → Det Nbar	*men* : N	*us* : Pro	*on* : P
Nbar → N	*saw* : N	*you* : Pro	*see* : V
Nbar → Nbar PP	*telescope* : N	*he* : Pro	*sees* : V
VP → V	*mountain* : N	*him* : Pro	*saw* : V
VP → V NP	*the* : Det	*she* : Pro	*die* : V
VP → VP PP	*this* : Det	*her* : Pro	*dies* : V
PP → P NP	*these* : Det	*it* : Pro	*died* : V

Where do grammars and lexicons come from? As usual in AI, there's knowledge engineering or there's machine learning. A human might manually write the grammar rules and lexical entries. For the lexical entries, of course, they are helped enormously by the fact that many conventional dictionaries are now available on-line; lexical entries can be extracted from these.

Machine learning has, in my view, been less successful. Much as with the tagging example earlier, we would require humans to manually draw parse trees for numerous sentences. These would then be the training set for a machine learning algorithm. The problems are: the manual parsing has proved unreliable; the rules learned are often only probabilistic ones (which may be a good or a bad thing depending on your point of view), and learning recursive rules has been problematic.

The module of an NLP system that uses the grammar and lexicon is known as the *parser*. The parser determines whether the input string is syntactically well-formed and, if it is, it determines the phrase-structure of the sentence. This structure will be used in some fashion to drive the process of computing the meaning of the sentence.

Note the difference between a parser and a tagger. Taggers simply assign labels, such as noun, verb, preposition, etc., to words. Parsers will do this too. But they will also say whether the string is grammatical or not, and they will also find the phrase-structure(s) of the string (the parse tree(s)).

4 Semantics

There are at least three types of knowledge of meaning: *semantics*, *pragmatics*, and *world knowledge*, but the boundaries between the three is contentious.

The division that we shall draw here is that semantics is concerned with *context-independent linguistic* meaning, pragmatics is concerned with *context-dependent linguistic* meaning, and world knowledge is concerned with *non-linguistic* meaning.

Semantics deals with the meanings of words and how those meanings combine together (in ways determined by the syntactic analyses) to give the meanings of phrases and sentences.

Consider example sentence (3):

(3) *"I spoke to you yesterday."*

Clearly the meaning of (3) changes according to who the speaker is, who the addressee is and the day on which it is uttered (so that we can know to whom or to what *"I"*, *"you"* and *"yesterday"* are referring). But, semantics tells us the context-independent meaning; it tells us that (3) is true if the speaker said something to the addressee on the day before the day on which utterance of (3) is made.

Current theories of semantics assume that meaning is assigned *compositionally*, that is to say, that the meaning of a linguistic expression is a function of the meaning of its parts. For example, to obtain the semantics of an S, we must combine the semantics of its constituent NP and VP, and the semantics of these will, in turn, be composed of the semantics of their constituents. It follows that we can associate semantic rules with grammar rules. Grammar rules say how an expression is made up of subexpressions; the corresponding semantic rule will say how the semantics of the expressions is made up of the semantics of its subexpressions. In most NLP systems, the semantics is represented explicitly in some logic.

The component of an NLP system that deals with semantic knowledge is the *semantic translator*. The semantic translator will compose fragments of logic to form wffs of logic. It will do this by using semantic rules, each such rule being associated with a grammar rule. So, in most systems, the parser 'drives' the semantic translator. Where the parser combines two or more words or phrases into a larger phrase, the semantic translator performs a corresponding computation for the semantics: the semantic meanings of the smaller units are combined to form the semantics of the larger unit. (You can see that this means that parse trees are not really needed.)

Bear in mind that some words in the lexicon will have more than one meaning. For example:

(4) *"The secretary couldn't produce letters on the old typewriter."*

The word *"letters"* could mean alphabetic characters or addressed documents. So even if there is only one parse tree, we may need to produce multiple wffs of logic.

5 Pragmatics

As we have already established, pragmatics characterises those aspects of linguistic meaning that are context-dependent: they vary from situation to situation. There are at least two aspects to pragmatics. The first aspect is to characterise how certain phrases *refer* to items in the context. The context, we have said, is the situation of use of the sentence, comprising who the speaker is, who the addressee is, the date, the time, the place, the shared cultural (and sub-cultural) knowledge, and salient parts of past conversations and the present conversation, etc.

Thus, (3), repeated here as (5):

(5) *"I spoke to you yesterday."*

if uttered by Ann on Wednesday when addressing Ben, will have its meaning 'fleshed out': *"I"* refers to Ann, *"you"* refers to Ben, and *"yesterday"* refers to some Tuesday.

Working out what is being referred to is not always easy. For example:

(6) *"Ann took her drink and sat by a pond. She drank it."*

Here, the referring expression *"it"* can refer either to Ann's drink or to the pond.

Pragmatics' second role is to work out the 'force' of the *speech act* (whether the speaker is stating something, asking a question, apologising, promising, etc.). This may not always be obvious. For example, sentence (7):

(7) *"Can you pass the salt?"*

looks like a question but, on most occasions, is intended as a request for action.

There are many other aspects of context-dependent meaning for which pragmatics must give an account. Here are a few examples:

- In certain contexts, utterances can be used non-literally, e.g. many phrases are intended metaphorically (*"We are losing the battle against unemployment"*) or idiomatically (*"He shot himself in the foot"*).
- In certain contexts, utterances can be intended to convey more than their literal meaning. For example:

  (8) Ann: *"Where's Col?"*
  Ben: *"Well, the pubs are open."*

  Ben's utterance, in addition to its literal meaning, licenses a plausible inference that Col is down the pub.

Theories of pragmatics are still relatively new and therefore few NLP systems carry out sophisticated pragmatic processing.

6 World Knowledge

Obviously linguistic knowledge (i.e. knowledge of what words there are, how they fit together to make sentences, what they denote, what their conditions of use are, etc.) is essential to us. But one of the things that makes NLP difficult is that we must make use not just of linguistic knowledge but also of *world knowledge* (background knowledge/ commonsense knowledge). World knowledge is our general knowledge about people, books, elephants, lectures, heroism, and so on, and our specific knowledge of individual people, books, elephants, lectures and acts of heroism.

A proper understanding of natural language utterances requires the use of non-linguistic knowledge, as much as it requires the use of linguistic knowledge. The following story, for example, cannot be fully understood without invoking large amounts of world knowledge to link the events described by the two utterances:

(9) *"Ann was condemned to death by the court. She had pushed her husband out of the window of their tenth floor apartment."*

Furthermore, it is most often world knowledge that enables a hearer to disambiguate ambiguous utterances. For example, in both of the following utterances the word *"they"* can refer either to the fascists or to the council. However, our world knowledge tells us that in (10) the more likely disambiguation is that *"they"* refers to the fascists since fascists generally promote violence and councils do not; and, in (11) the more likely disambiguation is that *"they"* refers to the council because councils rather than fascists are more likely to fear violence:

(10) *"The council refused to give the fascists a permit for a demonstration because they advocated violence."*

(11) *"The council refused to give the fascists a permit for a demonstration because they feared violence."*

Exercise

Use the grammar and lexicon given in section 3 to draw parse trees for the following strings. If you think that a string has more than one parse tree according to the grammar, draw them all. If you think that a string has no parse tree according to the grammar, then say so.

1. *"Ben died on the mountain."*
2. *"The mountain dies on Ben."*
3. *"I died he."*
4. *"The man with the telescope on the mountain died."*