CHAPTER

4

Natural Language Processing

Natural Language Processing:

Interaction among intelligent software agents is mainly realized by means of communication. Communication may vary from simple forms to sophisticated ones, as the one based on speech act theory. A simple form of communication is that restricted to simple signals, with fixed interpretations. Another form of communication is by message passing between agents.

When people communicate, they perform more than just exchanging messages with a restricted syntax and a given protocol, as in distributed systems. A more elaborate type of communication is communication based on the *speech act theory*. In such an approach, interaction among agents take place at least at two levels: one corresponding to the informational content of the message and the other corresponding to the intention of the communicated message. If interaction among agents is performed by means of message passing, each agent must be able to deduce the intention of the sender regarding the sent message.

In a speech act, there is a distinction between

- the locutionary act = uttering of words and sentences with a meaning;
- the illocutionary act = intent of utterance, e.g., request, inform, order, etc.;
- the prelocutionary act = the desired result of utterance, e.g., convince, insult, make do, etc.

One of the well known examples of interaction language among intelligent agents, based on speech act theory is the KQML (Knowledge Query and Manipulation Language) language proposed by ARPA Knowledge Sharing Effort in 1992. KQML uses the KIF (Knowledge Interchange Format) language to describe the content of a message. KIF is an ASCII representation of first order predicate logic using a LISP-like syntax.

Using a formal language, such as KIF or other, makes communication simpler. Things become more difficult if the communication language is one of the natural languages spoken by people in the world.

Communication between artificial intelligent agents refers to both the speaker's processes and to the hearer's processes. The speaker's processes are: intention, generation, and synthesis. NL generation is a distinct part of NL processing while NL synthesis refers to the synthesis of spoken words.

The hearer's processes are: perception, analysis, disambiguation, and incorporation. If the perception refers to spoken utterances then this is speech recognition. If it refers to hand written utterances then this is recognition of hand writing. Neural networks have been successfully used to both speech recognition and to hand writing recognition.

The analysis, disambiguation and incorporation form natural language understanding and are relying on the assumption that the words of the sentence are known. Many times, recognition of individual words may be driven by the sentence structure, so perception and analysis interact, as well as analysis, disambiguation, and incorporation.

Science fiction has been too optimistic in estimating progress towards NLP. An example from over 30 years ago, "2001: A Space Odyssey", made predictions for computers used at the turn of the century. One of these, HAL, was able to have meaningful dialogues with the astronauts. Speech recognition and understanding together with psychological advice were packaged into a friendly mentor. Interestingly, whilst chatting with HAL, the astronauts would use a clip board and pen to monitor the state of the space ship. Today microprocessors, invented 5 years after the book was written, monitor and control car engines thousands of times a second and yet our PC cannot understand a sentence such as "Can you delete this file?"

Early AI programs restricted their focus to microworlds, limited applications that require minimal domain knowledge. One of the earliest programs to take this approach was Terry Winograd's SHRDLU (Winograd, 1972), which could converse about a blocks world consisting of differently shaped and colored blocks and a hand for moving them about.

Winograd said on SHRDLU:

"Our program does not operate by first parsing a sentence, then doing semantic analysis, and finally using deduction to produce a response. These three activities go on concurrently throughout the understanding of a sentence. As an example, for the sentence "Put the blue pyramid on the block in the box", the parser first comes up with "blue pyramid on the block" as a candidate for a noun group. At this point, semantic analysis is done, and since "the' is definite, a check is made in the data base for the object being referred to. When no object is found, the parsing is redirected to find the noun group "the blue pyramid." It will then go on to find "on the block in the box" as a single phrase indicating a location. Thus, there is a continuing interplay between the different sorts of analysis, with the results of one affecting the others"

Types of communicating agents

1 Agents that share a common internal communication language

they can communicate without any external language at all

Problems:

- easy to agree on static symbols but difficult to agree on dynamic symbols
- need a renaming policy
- need a way to relate symbols introduced by different agents
- what to communicate and what new things the other agents have found out

2 Agents that make no assumption about each other's internal language

– they share a common communication language

Therefore, the agents have to deal with two languages:

- the communication language (external)
- the internal representation language

Although the specific organization of natural language understanding programs varies with different approaches and methods, all of them must translate the original sentence into an internal representation of its meaning. Some representations commonly used are: logic-based representations, conceptual graphs, conceptual dependencies, frames.

Defining the language (subset of English)

Lexicon – list of allowable vocabulary words grouped in categories (parts of speech)

Categories:

- open classes = words are added to the category all the time (NL is dynamic it constantly evolves);
- closed classes = small number of words, generally it is not expected that other words will be added.

Lexicon

Noun → stench | breeze | wumpus ..

Verb → is | see | smell ..

Adjective → right | left | smelly …

Adverb → here | there | ahead …

Pronoun → me | you | I | it

RelPronoun → that | who

Name → John | Mary

Article → the | a | an

Preposition → to | in | on

Conjunction → and | or | but

Grammar

S → NP VP | S Conjunction S

NP → Pronoun | Noun | Article Noun | NP PP | NP RelClause

VP → Verb | VP NP | VP Adjective | VP PP | VP Adverb

PP → Preposition NP

RelClause → RelPronoun VP

Syntactic analysis

Parsing is the problem of constructing a derivation tree for an input string from a formal definition of a grammar. Parsing algorithms may be divided into two classes:

Top down parsing – begin with the top-level sentence symbol and attempt to build a tree whose leaves match the target sentence's words (the terminals)

"John hit the ball"

1. S
2. S → NP, VP
3. S → Noun, VP
4. S → John, Verb, NP
5. S → John, hit, NP
6. S → John, hit, Article, Noun
7. S → John, hit, the, Noun
8. S → John, hit, the, ball

- Better if many alternative terminal symbols for each word
- Worse if many alternative rules for a phrase

Bottom-up parsing – start with the words in the sentence (the terminals) and attempt to find a series of reductions that yield the sentence symbol.

1. John, hit, the, ball
2. Noun, hit, the, ball
3. Noun, Verb, the, ball
4. Noun, Verb, Article, ball
5. Noun, Verb, Article, Noun
6. NP, Verb, Article, Noun
7. NP, Verb, NP
8. NP, VP
9. S

- Better if many alternative rules for a phrase
- Worse if many alternative terminal symbols for each word

Depth First parsing

- Try rules one at a time and back track if you get stuck
- Easier to program
- Less memory required
- Good if parse tree is deep

Breadth First parsing

- Try all rules at the same time
- Can be faster
- Order of rules is not important
- Good if tree is flat

Definite Clause Grammars (DCG)

A grammar written with logical sentences is called a logical grammar.

DCG rules may be written as PROLOG clauses and the PROLOG interpreter is used to perform top-down, depth-first parsing.

BNF	FOPL	PROLOG
S → NP VP NP → Noun Noun → stench Noun → wumpus VP → Verb Verb → smells Verb → kills	NP(s1) ∧ VP(s2) ⇒ S(append(s1,s2)) Noun(s) ⇒ NP(s) Verb(s) ⇒ VP(s) (s = "stench" ∨ s = "wumpus") ⇒ Noun(s) (v = "smells" ∨ v = "kills") ⇒ Verb(v)	sentence([S1, S2]) :- np(S1), vp(S2). np(S):- noun(S). vp(S):- verb(S). noun(stench). noun(wumpus). verb(smells). verb(kills). ?- sentence([wumpus, smells]). ?-sentence([S1, S2]).

Augmenting the DCG

- Nonterminals can be augmented with extra arguments, e.g., to verify grammatical correctness or attach semantics
- Add logical tests in the grammar rule – the rule fires only if the tests are true

▸ ***Add one extra argument for the semantics – see also semantic analysis further on***

DCG	FOPL	PROLOG
S(sem) → NP(sem1) VP(sem2) {compose(sem1, sem2, sem)}	NP(s1, sem1) ∧ VP(s2, sem2) ⇒ S(append(s1, s2)), compose(sem1, sem2, sem)	See later on

Compositional semantics

▸ Verify grammatical correct sentences

Problem: the previous grammar will generate sentences that are not grammatically correct.

- NL is not a context free language
- We must deal with
- cases
- agreement between subject and main verb in the sentence (predicate)
- verb subcategorization: the complements that a verb can accept

CASES

Nominative case (subjective case) + agreement

I take the bus	Je prends l'autobus
You take the bus	Tu prends l'autobus
He takes the bus	Il prend l'autobus

Accusative case (objective case)

He gives me the book	Il me donne le livre

Dative case

You are talking to me	Il parle avec moi

Solution to cases: new categories, e.g. NPS, NPO - not very efficient, too many rules

▸ Augment the DCG with a new parameter to describe the case

S → NP(Subjective) VP

NP(case) → Pronoun (case) | Noun | Article Noun // I

VP → VP NP(Objective) // believe him

VP → VP PP // turn to the right

VP → VP Adjective

VP → Verb

PP → Preposition NP(Objective)

Pronoun(Subjective) → I | you | he | she

Pronoun(Objective) → me | you | him | her

▸ Augment the DCG with a new parameter to describe the verb subcategorization

Verb subcategories – specify which verb can be followed by which other categories each verb has a list of complements

Augment VP to take a subcategorization argument

VP(subcat) → {subcat = np} VP(np) NP(Objective)

| {subcat = adj} VP(adj) Adjective

| {subcat = pp} VP (pp) PP

| Verb

change S so that it has a VP with subcategories

S → NP(Subjective) VP(subcat)

Add adjuncts to VP – verb phrases that may follow any verb, regardless of the subcategory

VP(subcat) → VP(subcat) PP
| VP(subcat) Adverb *I smell the wumpus now*

Resulting augmented DCG

S → NP(Subjective) VP(subcat)
NP(case) → Pronoun (case) | Noun | Article Noun
Pronoun(Subjective) → I | you | he | she
Pronoun(Objective) → me | you | him | her

VP(subcat) → {subcat = np} VP(np) NP(Objective)

| {subcat = adj} VP(adj) Adjective

| {subcat = pp} VP (pp) PP

| Verb

| VP(subcat) PP

| VP(subcat) Adverb

Note: top down parsing may be problematic with this DCG as it is left recursive.

We can try to implement it in Prolog but we must take care at left recursive clauses.

We can also use bottom-up parsing.

PROLOG

```
sentence(S) :- np(S1,subjective), vp(S2, Subcat), append(S1, S2, S).
np([S], Case) :- (Case=subjective; Case=objective),pronoun(S, Case).
np([S], _ ) :- noun(S).
np([S1, S2], _ ) :- article(S1), noun(S2).
pronoun(i, subjective).
pronoun(you, _ ).
pronoun(he, subjective).
pronoun(she, subjective).
pronoun(me, objective).
pronoun(him, objective).
pronoun(her, objective).
noun(ball).
noun(stick).
article(a).
article(the).
```

But dangerous to translate

VP(subcat) → VP(subcat) … in Prolog because infinite recursion

Solution

```
vp(S,Subcat):- Subcat=np,vp1(S1,np),np(S2,objective),append(S1,S2,S).
vp1([S],np):-verb(S).
verb(give).
verb(make).
```

vp(S,Subcat) :- vp1(S1, Subcat), pp(S2), append(S1, S2, S).

Note: There are still some adjustments to make in order for the above Prolog program to work according to correct rules of English

Semantic analysis

Compositional semantics

The dog has legs. (dog *parts* legs)
The ball is yellow. (ball *color* yellow)
The ball is red. (ball *color* red)
The dog bites. (dog *action* bytes)

Augment DCG with semantic interpretation

PROLOG (without case and subcategories)

sentence(S, Sem) :- np(S1, Sem1), vp(S2, Sem2), append(S1, S2, S),
Sem = [Sem1 | Sem2].

np([S1, S2], Sem) :- article(S1), noun(S2, Sem).

vp([S], Sem) :- verb(S, Sem1), Sem = [action, Sem1].

vp([S1, S2], Sem) :- verb(S1), adjective(S2, color, Sem1),

Sem = [property, Sem1].

vp([S1, S2], Sem) :- verb(S1), noun(S2, Sem1), Sem = [parts, Sem1].

But NL does not have a compositional semantics for the general case. Then the semantic interpretation is responsible for getting all possible interpretations, and disambiguation is responsible for choosing the best one. Disambiguation is done starting from the pragmatic interpretation of the sentence.

Pragmatic interpretation

- Complete the semantic interpretation by adding information about the current situation
- How the language is used and its effects on the listener. Pragmatics will tell why it is not appropriate to answer "Yes" to the question "Do you know what time it is?"

- Indexical = phrases that refer directly to the current situation

I am in Paris today

- Anaphora = the occurrence of phrases referring to objects that have been mentioned previously

Mary was sleepy. She went to bed.

The ball hit the house. It broke a window.

John meet Mary at the bus station. They went to a restaurant.

Ambiguity

- Lexical ambiguity

A clear sky A clear profit
The way is clear John is clear It is clear that …

- Syntactic ambiguity

I saw the Statue of liberty flying over New York
Salespeople sold the dog biscuits
I saw John in a restaurant with a telescope
Time flies like an arrow

- Referential ambiguity

Block A is on block B and it is not clear
John met Mary and Tom. They went to a restaurant.

- Pragmatic ambiguity

I'll meet you next Friday
I am in the room with John

Disambiguation

- the world model
- the mental model
- the language model
- the acoustic model

Other approaches to NL analysis

Augmented Transition Networks

A transition network (TR) represents a grammar as a set of finite-state machines or transition networks. each network corresponds to a single nonterminal in the grammar. Arcs in the network are labeled with either terminal or nonterminal symbols. Each path trough the network, from the start state to the final state, corresponds to some rule for that nonterminal; the sequence of arc labels on the path is the sequence of symbols on the right hand side of the rule. When there is more than one rule for a nonterminal, the corresponding network has multiple paths from the start node to the end node. Finding a successful transition through the network for a nonterminal corresponds to the replacement of that nonterminal by the right hand side of a grammar rule. To parse a sentence, a transition network parser must find a transition trough the sentence network.
Augmented transition networks (ATN) extend TR by allowing procedures to be attached to the arcs of the networks. An ATN parser executes these attached procedures when it traverses the arcs. The procedures may assign values to grammatical features and perform tests, causing a transition to fail if certain conditions (such as number agreement) are not met. These procedures also construct a parse tree, which is used to generate an internal semantic representation of the sentence's meaning. Both terminals and nonterminals are represented as categories (e.g., verb, noun phrase) with attached features. For example, a word is described using its morphological root, along with features for its part of speech, number, person, etc.

One problem with ATNs (especially in speech recognition) is that their procedurality limits their effectiveness. It is sometimes better to parse a sentence based on clearly recognized parts when earlier parts are still unknown.

Recursive transition networks

Recursive transition networks (RTN) generalize finite state automata by allowing nondeterministic transitions between two states to be taken via a recursive jump to a start state. ATNs are RTNs that also have a finite set of registers and actions that can set registers to input words, corresponding lexical entries, or to some function of the content of other registers. Recursive calls to the network can pass values back to the calling level.

Montague semantics

Montague semantics, also known as compositional semantics is based on the following mechanism. For every step in the syntactic parsing process, there is a corresponding step in semantic interpretation. Each time syntactic constituents are combined to form a larger syntactic unit, their corresponding semantic interpretation can be combined to form a larger semantic unit. The clearest application of this idea is the work of Montague. The compositional approach to defining semantic interpretation has proved to be a very powerful idea. Unfortunately, there are some linguistic constructions (such as quantification) that cannot be accounted for in this way.

SOLVED PROBLEMS

1. Natural Language Processing (NLP) is field of

(a) Computer Science (b) Artificial Intelligence

(c) Linguistics (d) All of the mentioned

Soln. **Correct option is (d)**

2. One of the main challenge/s of NLP Is _____

(a) Handling Ambiguity of Sentences (b) Handling Tokenization

(c) Handling POS-Tagging (d) All of the mentioned

Soln. There are enormous ambiguity exists when processing natural language.

Correct option is (a)

3. Choose form the following areas where NLP can be benifited.

(a) Automatic Text Summarization

(b) Automatic Question-Answering Systems

(c) Information Retrieval

(d) All of these

Soln. **Correct option is (d)**

4. The major tasks of NLP includes

(a) Automatic Summarization (b) Discourse Analysis

(c) Machine Translation (d) All of these

Soln. There is even bigger list of tasks of NLP.

Correct option is (d)

5. Coreference Resolution is

(a) Anaphora Resolution

(b) Given a sentence or larger chunk of text, determine which words ("mentions") refer to the same objects ("entities")

(c) All of these

(d) None of these

Soln. Anaphora resolution is a specific type of coreference resolution.

Correct option is (b)

6. Machine Translation

(a) Converts one human language to another

(b) Converts human language to machine language

(c) Converts any human language to English

(d) Converts Machine language to human language

Soln. The best known example of machine translation is google translator.

Correct option is (a)

7. Morphological Segmentation

(a) Does Discourse Analysis

(b) Separate words into individual morphemes and identify the class of the morphemes

(c) Is an extension of propositional logic

(d) None of these

Soln. **Correct option is (b)**

8. Parts-of-Speech tagging determines

(a) part-of-speech for each word dynamically as per meaning of the sentence

(b) part-of-speech for each word dynamically as per sentence structure

(c) all part-of-speech for a specific word given as input

(d) all of these

Soln. A Bayesian network provides a complete description of the domain.

Correct option is (d)

9. Many words have more than one meaning; we have to select the meaning which makes the most sense in context. This can be resolved by

(a) Fuzzy Logic (b) Word Sense Disambiguation

(c) Shallow Semantic Analysis (d) All of these

Soln. Shallow Semantic Analysis doesn't cover word sense disambiguation.

Correct option is (b)

10. Given a sound clip of a person or people speaking, determine the textual representation of the speech.

(a) Text-to-speech (b) Speech-to-text

(c) All of these (d) None of these

Soln. NLP is required to linguistic analysis.

Correct option is (b)

11. In linguistic morphology, _____ is the process for reducing inflected words to their root form.

(a) Rooting (b) Stemming

(c) Text-Proofing (d) Both Rooting & Stemming

Correct option is (b)