Artificial Intelligence

CSE-401

UTKARSH GUPTA

A2305217557

Q1. Describe the process of syntactic analysis in NLP. How can you say that the ultimate goal of NLP is to help computers understand language? Explain with facts.

Ans. Syntactic analysis or parsing or syntax analysis is the third phase of NLP. The purpose of this phase is to draw exact meaning, or you can say dictionary meaning from the text. Syntax analysis checks the text for meaningfulness comparing to the rules of formal grammar. For example, a sentence like "hot ice-cream" would be rejected by a semantic analyzer. In this sense, syntactic analysis or parsing may be defined as the process of analyzing the strings of symbols in natural language conforming to the rules of formal grammar. The origin of the word 'parsing' is from the Latin word 'pars' which means 'part'.

Natural language processing (NLP) is an area of computer science and artificial intelligence concerned with the interaction between computers and humans in natural language. The ultimate goal of NLP is to help computers understand language as well as we do. It is the driving force behind things like virtual assistants, speech recognition, sentiment analysis, automatic text summarization, machine translation and much more. In this

post, we'll cover the basics of natural language processing, dive into some of its techniques and also learn how NLP has benefited recent advances in deep learning.
Central to deep learning and natural language is "word meaning," where a word and especially its meaning are represented as a vector of real numbers. With these vectors that represent words, we are placing words in a high-dimensional space. The interesting thing about this is that the words, which are represented by vectors, will act as a semantic space. This simply means the words that are similar and have a similar meaning tend to cluster together in this high-dimensional vector space.

Q2. Define is a conceptual dependency? How is it different from the frame? Give conceptual
dependency representation for:
a) Satish gave chocolate to Rajesh.
b) Krishna goes to the temple by car.

Ans. Conceptual Dependency originally developed to represent knowledge acquired from natural language input. A set of conceptual transitions then act on this representation, e.g. an ATRANS is used to represent a transfer such as "give" or "take" while a PTRANS is used to act on locations such as "move" or "go". An MTRANS represents mental acts such as "tell", etc.

The goals of this theory

are:

1. To help in the drawing of inference from sentences.
2. To be independent of the words used in the original input.

That is to say: For any 2 (or more) sentences that are identical in meaning there should be only one representation of that meaning.
It has been used by many programs that portend to understand English (MARGIE, SAM, PAM). CD developed by Schank et al. as were the previous examples.
Frames and scripts, as knowledge representation schemes take into account context and relationships. These techniques also provide a useful formulation for representing more complex structures such as objects, scenes and multiple-sentence stories. One of the key ideas of the script approach is to reduce a sentence or story to a set of semantic primitives using a formalism called conceptual dependency (CD).
Conceptual dependency (CD) is a theory of natural language processing that mainly deals with the representation of the semantics of a language.

Q4. Describe Scripts' representation. Write the various component of Script representation. Represent a complete script for bank robbery.

Ans A script is a structured representation describing a stereotyped sequence of events in a particular context.
Scripts are used in natural language understanding systems to organize a knowledge base in terms of the situations that the system should

understand. Scripts use a frame-like structure to represent the commonly occurring experience like going to the movies eating in a restaurant, shopping in a supermarket, or visiting an ophthalmologist.
Thus, a script is a structure that prescribes a set of circumstances that could be expected to follow on from one another.
Scripts are beneficial because:
1. Events tend to occur in known runs or patterns.
2. A casual relationship between events exist.
3. An entry condition exists which allows an event to take place.
4. Prerequisites exist upon events taking place.

Components of a script

The components of a script include:
1. Entry condition: These are basic condition which must be fulfilled before events in the script can occur.
2. Results: Condition that will be true after events in script occurred.
3. Props: Slots representing objects involved in events
4. Roles: These are the actions that the individual participants perform.
5. Track: Variations on the script. Different tracks may share components of the same scripts.
6. Scenes: The sequence of events that occur.

Q5. Write a finite-state automaton for a dialogue manager for checking your bank balance

and
withdrawing money at an automated teller machine.
Ans
a) Definition. Automated teller machine operator means any person that operates an automated teller machine at which a consumer initiates an electronic fund transfer or a balance inquiry and that does not hold the account to or from which the transfer is made, or about which an inquiry is made.
(b) General. An automated teller machine operator that imposes a fee on a consumer for initiating an electronic fund transfer or a balance inquiry shall:
(1) Provide notice that a fee will be imposed for providing electronic fund transfer services or a balance inquiry, and
(2) Disclose the amount of the fee.
(c) Notice requirement. To meet the requirements of paragraph (b) of this section, an automated teller machine operators must comply with the following:

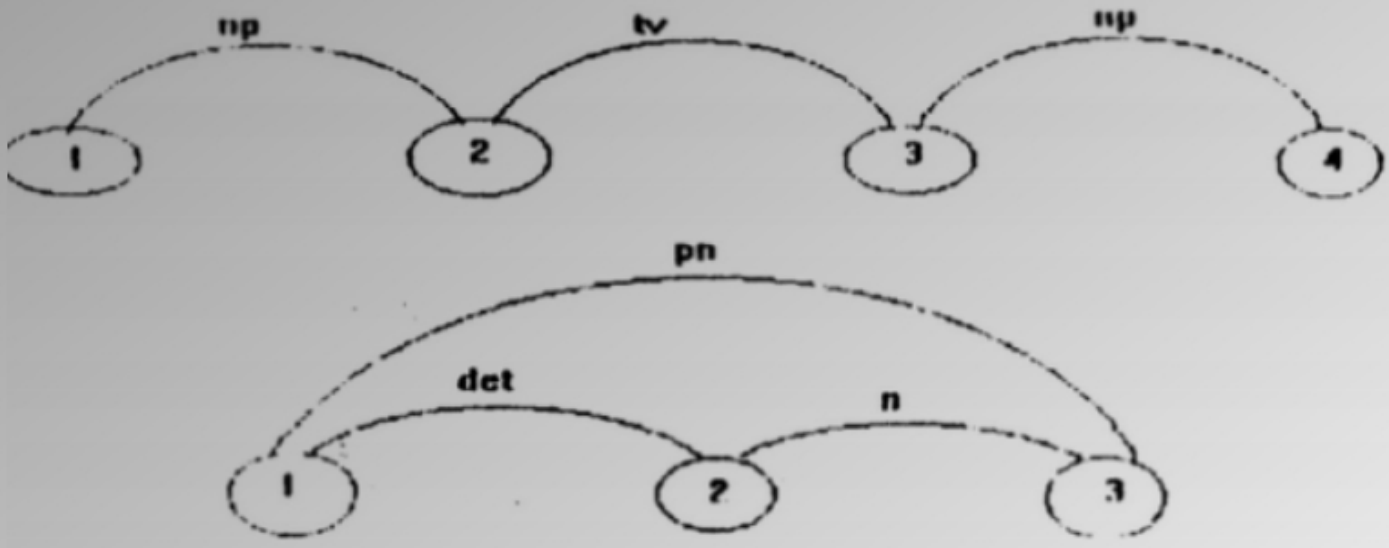
(1) On the machine. Post in a prominent and conspicuous location on or at the automated teller machine a notice that:
(i) A fee will be imposed for providing electronic fund transfer services or for a balance inquiry, or (ii) A fee may be imposed for providing electronic fund transfer services or for a balance inquiry, but the notice in this paragraph
(c)(1)(ii) may be substituted for the notice in paragraph

(c)(1)(i) only if there are circumstances under which a fee will not be imposed for such services; and (2) Screen or paper notice. Provide the notice required by paragraphs
(b)(1) and (b)(2) of this section either by showing it on the screen of the automated teller machine or by providing it on paper, before the consumer is committed to paying a fee.
(d) Temporary exemption. Through December 31, 2004, the notice requirement in paragraph (c)(2) of this section does not apply to any automated teller machine that lacks the technical capability to provide such information.
(e) Imposition of the fee. An automated teller machine operator may impose a fee on a consumer for initiating an electronic fund transfer or a balance inquiry only if
(1) The consumer is provided the notices required under paragraph (c) of this section, and
(2) The consumer elects to continue the transaction or inquiry after receiving such notices.

Q6 Why transition network is important for NLP? Differentiate Transition Network, Recursive Transition Network, and Augmented Transition Network (ATN) by taking a suitable example(s).

Ans. The basic assumption of FSTNs is that humans produce sentences one word at a time. Of all the models of representing formal and natural

language structures, Finite State Transition Networks (FSTNs) are the simplest and based on the application of directed graphs. An FSTN consists of a number of nodes and labeled arcs. When we traverse a sentence, each node represents a state and arcs represent rules or test conditions required to make transition from one state to the next state. Thus the total path reflects a valid sequence of words to accept a grammatically correct sentence. The parser proceeds from left-to-right and imposes constraint rules that limit the choice of words that can be accepted for use. at. any particular state in the production of a sentence. For example, if the first, word is a determiner the next word should be either an adjective or a noun. In other words, a sentence analysis is viewed as a transition through a series of states.



There are generally two types of transition networks like

1. Recursive Transition networks (RTN)
2. Augmented Transition networks

(ATN)

Basically, RTNs are just like FSTNs except that they have subnetworks. It is quite practical for an arc to point to a subnetwork to be traversed, instead of a specific word. That is, if we have a commonly used bunch of arcs, we can express the abstraction by making it into a self-contained subnetwork with specific name. The subnetworks can be referred each time in the network where ever they appear. This naturally allows building large networks in a modular way. RTNs also allow us to represent efficiently some of the recursive structures in natural language.

The FSTNs and RTNs are capable of either rejecting or accepting a sentence depending on the syntax of a language. However, a language understand- Ch.3: NLP Tools & Techniques 107 ng system should be able to build structures that may be used for further analysis, like subject-verb agreement and for analysis of mood, tense, etc. Any RTN which allows additional tests and stores information on the labels, can be called an Augmented Transition Network (ATN). The main augmentation feature of ATN is its capability to store information in registers so that further tests can be carried out. ATNs provide registers for each subject like NPs, VPs and PPs. At the end of parsing the contents of registers are grouped to form a valid sentence structure, until then ATNs keep on trying alternative sentence structure.

Q7 A sentence can easily have more than one parse tree that is consistent with a given CFG. How do CFGs handle parsing

ambiguity?

Ans. PCFG parsers resolve ambiguity by preferring constituents (and parse trees) with the highest probability. Consider the following PCFG for problems (b)-(e). production rule probability

production rule	probability
S → VP	1.0
VP → Verb NP	0.7
VP → Verb NP PP	0.3
NP → NP PP	0.3
NP → Det Noun	0.7
PP → Prep Noun	1.0
Det → the	0.1
Verb → Cut | Ask | Find| ...	0.1
Prep → with | in | ...	0.1

Q8. Describe the use of context-free Grammar in NLP. Explain the concept of sentence generation and translation with reference of CFG.

Ans. A context-free grammar (CFG) is a list of rules that define the set of all well-formed sentences in a language. Each rule has a left-hand side, which identifies a syntactic category, and a right-hand side, which defines its alternative component parts, reading from left to right.
E.g., the rule s --> np vp means that "a sentence is defined as a noun

phrase followed by a verb phrase." Figure 1 shows a simple CFG that describes the sentences from a small subset of English.

G =

T = {that, this, a, the, man, book, flight, meal, include, read, does}

N = {S, NP, NOM, VP, Det, Noun, Verb, Aux}

S = S

R = {

S → NP VP Det → that | this | a | the

S → Aux NP VP Noun → book | flight | meal | man

S → VP Verb → book | include | read

NP → Det NOM Aux → does

NOM → Noun

NOM → Noun NOM

VP → Verb

VP → Verb NP }

Q9. Explain the concept of Augmented Transition Network (ATN). Explain it with the help of an example and with a suitable graph/picture. Justify the statement "ATMs build on the idea of using finite state machines to parse sentences"

Ans. An augmented transition network (ATN) is a type of graph structure used in the operational definition of formal languages, used especially in parsing relatively complex natural languages and having wide application in artificial intelligence. An ATN can, theoretically, analyze the structure

of any sentence, however complicated. ATNs build on the idea of using finite state machines to parse sentences. ATNs have states to mark the progress in processing a string. Transitions correspond to individual words or phrases from a syntactic type. W. A. Woods in "Transition Network Grammars for Natural Language Analysis" claims that by adding a recursive mechanism to a finite state model, parsing can be achieved much more efficiently. Instead of building an automaton for a particular sentence, a collection of transition graphs are built. A grammatically correct sentence is parsed by reaching a final state in any state graph. Transitions between these graphs are simply subroutine calls from one state to any initial state on any graph in the network. A sentence is determined to be grammatically correct if a final state is reached by the last word in the sentence.

Q10. How can you say that semantic analysis makes sure that declarations and statements of The program is semantically correct? How semantic analysis and NLP are related together? Explain.

Ans. Semantic Analysis is the third phase of Compiler. Semantic Analysis makes sure that declarations and statements of the program are semantically correct. It is a collection of procedures which is called by parser as and when required by the grammar. Both syntax trees of the previous phase and symbol table are used to check the consistency of the given code. Type checking is an important part of the semantic analysis where the compiler makes sure that each operator has matching

operands.

The semantic analysis of natural language content starts by reading all of the words in content to capture the real meaning of any text. It identifies the text elements and assigns them to their logical and grammatical role. It analyzes context in the surrounding text and it analyzes the text structure to accurately disambiguate the proper meaning of words that have more than one definition.

Semantic Analyzer:

It uses a syntax tree and a symbol table to check whether the given program is semantically consistent with language definition. It gathers type information and stores it in either a syntax tree or a symbol table. This type of information is subsequently used by the compiler during intermediate-code generation.

Semantic Errors:

>> Errors recognized by the semantic analyzer are as follows:

1. Type mismatch
2. Undeclared variables
3. Reserved identifier misuse

>>Functions of Semantic Analysis:

1. Type Checking -

Ensures that data types are used in a way consistent with their definition.

2. Label Checking -

A program should contain labels references.

3. Flow Control Check