NLP DIGITAL NOTES

UNIT – I:

Finding the Structure of Words: Words and Their Components, Issues and Challenges, Morphological Models Finding the Structure of Documents Introduction, Methods, Complexity of the Approaches, Performances of the Approaches

Natural Language Processing: Natural Language Processing (NLP) is a cross disciplinary field of linguistics, computer science, and artificial intelligence. It is related to the interactions between digitalized computing devices and human language or precisely natural language.

The field of natural language processing deals with designing and programming digital computational devices (particularly computers) to process and analyse large amounts of natural language data.

Natural languages take different forms, such as writing, speech or signing. They are unrestricted from constructed and formal languages such as those used to program computers or to study logic. As a result of its natural language data is highly unstructured in nature. **For example**, NLP makes it possible for computers to read text, hear speech, interpret it, measure sentiment and determine which parts are important.

Spectrum of natural languages: The spectrum of natural languages is very wide. As per the linguistic science There are thousands of spoken languages in the world. These languages can be grouped together as members of a language family. There are three main language in the world o Indo-European (Includes English) o Sino-Tibetan (Includes Chinese) o Afro-Asiatic (Includes Arabic). Natural language is a complicated thing. In many languages, words are delimited in the orthography by whitespace and punctuation. Some languages use word forms that need not change much with the changing context. On the other hand, there are languages that are highly sensitive about the choice of word forms according to context.

For some of the languages, the context does not impact the gender of the noun, while some languages do not have the concept of gender. x Natural languages show structure (namely grammar) of different kinds and complexity. It consists of more elementary components whose co-occurrence in context refines the purpose when used in isolation, but they extend it further to meaningful relations between other components in the sentence. x As a result, understanding natural language in word blocks is not a viable approach. However, the first level understanding of a word is very important.

Word morphology in natural language or understand words in natural language. ‹ Words are the most indicative blocks of a natural sentence. However, they are tricky to define. This is primarily due to ambiguity and contextual meaning of words in sentences. Knowing how to work with words allows, the development of syntactic and semantic understanding. x The process of understanding words in any natural language involves morphology, word structure and its linguistic expression. o Morphology is the study the variable forms and functions of words, while syntax is concerned with the arrangement of words into phrases, clauses, and sentences. o Word structure constraints due to pronunciation are described by phonology, whereas conventions for writing constitute the orthography of a language. o The linguistic expression is its semantics, and etymology and lexicology cover especially the evolution of words and explain the semantic, morphological and other links among them. x Word structure constraints due to pronunciation are described by phonology, whereas conventions for writing constitute the orthography of a language.In this unit we would discuss morphological parsing. x The technique of discovering of word structure is called morphological parsing. It is used to o Identify words of distinct types in natural languages. o Model different internal structure of words in connection with the grammatical properties and lexical concepts.

1.1 Words and their Components: Components of words from NLP perspective. ‹

x Words are defined as the smallest linguistic units in a natural language that can form a complete utterance by themselves. x The minimal parts of words that deliver aspects of

meaning to them are called morphemes. x Depending on the means of communication, morphemes are o Spelled out via graphemes (**symbols of writing**) such as letters or characters - or o Realized through phonemes (the distinctive units of sound in spoken language).

Specific to a particular language the exact boundaries of separating words from morphemes and phrases are varied and it is specific to that language. Here is an example with nouns as a valid word in English used to understand the above concept.

Noun	Noun + s (plural)	Noun + s (possessive)	Pronunciation(both)
thrush	thrushes	thrush's	iz
toy	toys	toy's	z
block	blocks	block's	s

Fig. Q.4.1 : Example to separate words from morphemes

Tokens: Tokens are syntactic words. Let's consider a simple sentence given below. I don't want to buy this product. x In the above sentence for reasons of generality, linguists prefer to analyse don't as two syntactic words (do not), or tokens, each of which has its independent role and can be reverted to its normalized form. On the other hand, all other words in this sentence are treated as an independent single token. x In English, such tokenization and normalization may be applied to a limited set of cases. However, in other languages, these phenomena have to be treated in a less trivial manner.

Clitics in morphology: In morphology and syntax, a cliticis a morpheme that has syntactic characteristics of a word, but depends phonologically on another word or phrase. x It is syntactically independent but phonologically dependent - always attached to a host. x A clitic is pronounced like an affix, but plays a syntactic role at the phrase level. x In other words, clitics have the form of affixes, but the distribution of function words. x For example, the contracted forms of the auxiliary verbs in I'm and we've are clitics. x As a result, underlying lexical or syntactic.

In Sanskrit language the token Namaste is a Clitic formed due to Sandhi of two tokens < *namah + te*> (देवनागरीनमः + ते = नमस्ते).

The sandhi changed token
namah → namas(नमः→नमस्).

For example:

x Tokens behaving in this way can be found in various languages like Latin, Ancient Greek, Chinese, Japanese, Sanskrit, Czech, Tamil, Telugu, Estonian and many more. x In such languages, tokenization, also known as word segmentation, is the fundamental step of morphological analysis and a pre-requisite for most NLP.

Importance of lexemes as a linguistic form. OR Discuss the lexemes.

x In a natural language a word often denotes one linguistic form in the given context and

the concept behind the form and the set of alternative forms that can express it. Such sets are called lexemes or lexical items. They together form the lexicons of a language. x Lexemes can be divided by their behaviour into the lexical categories of verbs, nouns, adjectives, conjunctions, particles, or other parts of speech. x The citation form of a lexeme is also called its lemma. x The notion of the lexeme is central to morphology and it is the basis for defining other concepts in the morphology. For example, the difference between inflection and derivation can be stated in terms of lexemes: o Inflectional rules relate a lexeme to its forms. o When we convert a word into its other forms, such as turning the singular mouse into the plural mice or mouses, we say we inflect the lexeme. o Derivational rules relate a lexeme to another lexeme. o When we transform a lexeme into another one that is morphologically related, regardless of its lexical category, for instance, the nouns consumer and consumption are derived from the verb to consume.

Morphemes as the smallest meaningful unit in a language. Morphemes vs words

Ans.: x There are different opinions on whether and how to associate the properties of word forms with their structural components. These components are usually called segments or morphs. x The morphs that by themselves represent some aspect of the meaning of a word are called morphemes of some function. x A morpheme is the smallest meaningful unit in a language. x A morpheme is not identical to a word. The main difference between them is that a morpheme sometimes does not stand alone, but a word, by definition, always stands alone. x When a morpheme stands by itself, it is considered as a root because it has a meaning of its own (such as the morpheme dog). When it depends on another morpheme to express an idea, it is an affix because it has a grammatical function (such as the –s in dogs to indicate that it is plural) x Natural languages use different techniques by which morphs and morphemes are combined into word forms. The simplest morphological process concatenates morphs one by one. x For example, as in the word, mis-manage-ment-s, where manage is a free lexical morpheme and the other elements are morphemes adding some meaning to the whole word. x For example, in Korean language, many morphemes change their forms with the phonological context. Fig. Q.8.1 shows some Korean morphemes -ess-, -ass-, -yess- indicating past tense.

	concatenated		contracted		
(a)	보았-	po-ass-	봤-	pwass-	'have seen'
(b)	가지었-	ka.ci-ess-	가졌-	ka.cyess-	'have taken'
(c)	하였-	ha-yess-	했-	hayss-	'have done'
(d)	되었-	toy-ess-	됐-	twayss-	'have become'
(e)	놓았-	noh-ass-	놨-	nwass-	'have put'

Fig. Q.8.1 : Korean morphemes indicating past tense

Allomorphs   x The alternative forms of a morpheme are termed allomorphs.   x Allomorphs are variants of a morpheme that differ in pronunciation but are semantically identical. For   example, the English plural marker -(e)s of regular nouns can be pronounced /-s/ (bags), (bushes), depending on the final sound of the noun's plural form. **Short note on following terminologies:**

(a) Typology (b) Isolating, or analytic typology (c) Synthetic languages (d) Agglutinative languages (e) Fusional languages

(f) Nonlinear languages ‹

(a) Typology x Typology (or Morphological typology) is a way of classifying the languages in the world. It groups languages according to their common morphological structures. Typology organizes languages on the basis of how those languages form words by combining morphemes. x The typology that is based on quantitative relations between words, their morphemes, and their features as follows.

(b) Isolating, or analytic typology x These languages include no or relatively few words that has more than one morpheme. Examples are Chinese, Vietnamese, and Thai. x Analytic languages show a low ratio of morphemes to words, nearly one-to-one. x Sentences in analytic languages are composed of independent root morphemes. x Grammatical relations between words are expressed by separate words where they might otherwise be expressed by affixes, which are present to a minimal degree in such languages. x Some analytic tendencies are also found in languages like English and Afrikaans.

(c) Synthetic languages x These can combine more morphemes in one word and are further divided into agglutinative and fusional languages. x The morphemes may be distinguishable from the root, or they may not. They may be fused with it or among themselves.

x Word order is less important for these languages than it is for analytic languages, since individual words express the grammatical relations that would otherwise be indicated by syntax. x In addition, there tends to be a high degree agreement or cross-reference between different parts of the sentence. x Therefore, morphology in synthetic languages is more important than syntax. x Most Indo-European languages are moderately synthetic.

(d) Agglutinative languages: x These languages have morphemes associated with only a single function at a time. x Agglutinative languages have words containing several morphemes that are always clearly differentiable from one another. x Each morpheme represents only one grammatical meaning and the boundaries between those morphemes

are easily demarcated. x The bound morphemes are affixes and they may be individually identified. x Agglutinative languages tend to have a high number of morphemes per word, and their morphology is usually highly regular. x Agglutinative languages include Finnish, Hungarian, Turkish, Mongolian, Korean, Japanese, Indonesian, Tamil etc. R (e) Fusional languages x These languages are defined by their feature-per-morpheme ratio higher than other languages. x Morphemes in fusional languages are not readily distinguishable from the root or among themselves. x Several grammatical bits of meaning may be fused into one affix. x Morphemes may also be expressed by internal phonological changes in the root. x The Indo-European and Semitic languages are the most typically cited examples of fusional languages. x Examples of fusional Indo-European languages are: Kashmiri, Sanskrit, Pashto, New Indo-Aryan languages such as Punjabi, Hindustani, Bengali; Greek (classical and modern), Latin, Italian, French, Spanish, Portuguese, Romanian, Irish, German, Faroese, Icelandic, Albanian and all Balto-Slavic languages. Concatenative languages x These languages link morphs and morphemes one after another.

(f) Nonlinear languages x Nonlinear languages allow structural components to merge non-sequentially to apply tonal morphemes or change the consonantal or vocalic templates of words. x It is also called discontinuous morphology and introflection, is a form of word formation and inflection in which the root is modified and which does not involve stringing morphemes together sequentially.

x For example, in English, mostly plurals are usually formed by adding the suffix - s, certain words use nonconcatenative processes for their plural forms as foot o feet x Many irregular verbs form their past tenses, past participles or both in the same manner: freeze o froze o frozen x This specific form of nonconcatenative morphology is known as base modification or ablaut, a form in which part of the root undergoes a phonological change without necessarily adding new phonological material x For example the English stem song, results in the four distinct words as Sing → sang → sung → song.

1.2 Issues and Challenges

Importance of morphological parsing and modelling in NLP. ‹

x Morphological parsing helps to eliminate or improve the inconsistency of word forms. It is required to provide higher-level linguistic units whose lexical and morphological properties are explicit and well defined.

x Every Natural language inherently has some irregularity and ambiguity. Morphological parsing attempts to remove unnecessary irregularity and control ambiguity.

x Irregularities o In this context irregularity means existence of such forms and structures that are not described appropriately by a prototypical linguistic model. o Some irregularities can be understood by redesigning the model and improving its rules, but other lexically dependent irregularities often cannot be generalized.

x Ambiguity o Ambiguity is an inability in interpretation of expressions of language. o Accidental ambiguity and ambiguity due to lexemes with multiple senses, cause syncretism, or systematic ambiguity. x Morphological modelling also faces the problem of productivity and creativity in language. This gives birth to unconventional but perfectly meaningful new words or new senses to the language. x Because these newly coined words are not present in the lexical and morphological properties, such words will remain completely unparsed in morphological system. This unknown word problem is particularly severe in speech or writing. x The morphological modelling is unable to parse a word, that comes from an expected domain of the linguistic model. This happens mostly when special terms or foreign names are involved in the discourse or when multiple languages or dialects are mixed together.

Morphological irregularities in NLP: x The design principles of the morphological model are very important to control the irregularities in words.

x Morphological parsing is designed for generalization and abstraction of words to make the model simple and yet powerful.

x However, the immediate descriptions of given for a word may not be the final ones, due to o Inadequate accuracy description o Inappropriate complexity of morphological model o Need of improved formulations x Removal of morphological irregularities o A deeper study of the morphological processes is essential for mastering the whole morphological and phonological system. o Morphophonemic templates capture morphological processes. It is done by organizing stem patterns and generic affixes. o These templates are designed without any context-dependent variation of the affixes or ad hoc modification of the stems. o A very terse merge rules ensure that morphophonemic templates can be converted into exactly the surface forms namely, orthographic and phonological. o Applying the merge rules is independent of and irrespective of any grammatical parameters or information other than that contained in a template. o Thus, most morphological irregularities in the morphophonemic templates are successfully removed.

Morphological irregularities in any two natural languages. ‹

Morphological irregularities in Arabic o Morphophonemic templates can be used for

discovering the regularity of Arabic morphology where uniform structural operations apply to different kinds of stems. o Some irregularities are bound to particular lexemes or contexts, and cannot be accounted for by general rules. x Morphological irregularities in Korean o Korean irregular verbs provide examples of such irregularities. Korean shows exceptional constraints on the selection of grammatical morphemes. o Korean language features lexically dependent stem alternation.

Morphological irregularities in other Natural languages o It is hard to find irregular inflection in agglutinative languages: Two irregular verbs in Japanese, one in Finnish. o These languages are abundant with morphological alternations that are formalized by precise phonological rules.

Morphological ambiguity:

x Morphological ambiguity is the possibility that word forms be understood in multiple ways out of the context. x Words forms that look the same but have distinct functions or meaning are called homonyms. x Ambiguity is present in all aspects of morphological processing and language processing at large. x Morphological parsing cannot complete disambiguation of words in their context, but it can control the valid interpretations of a given word form. x Morphological Ambiguity in Korean o In Korean, homonyms are one of the most problematic objects in morphological analysis. This is because they prevail all around frequent lexical items. x Morphological Ambiguity in Arabic o Arabic has rich derivational and inflectional morphology. Because Arabic script usually does not encode short vowels and omits yet diacritical marks, its morphological ambiguity is considerably increased. In addition, Arabic orthography collapses certain word forms together.

o The problem of morphological disambiguation of Arabic encompasses Ÿ

→The resolution of the structural components of words Ÿ

→Actual morphosyntactic properties Ÿ

→Tokenization and normalization Ÿ

→Lemmatization, stemming Ÿ

→Diacritization

x Morphological ambiguity in Sanskrit o When inflected syntactic words are combined in an utterance, additional phonological and orthographic changes can take place. o In Sanskrit, one such euphony rule is known as external sandhi. Inverting sandhi during tokenization is usually nondeterministic as it can provide multiple solutions. x In any language, tokenization decisions may impose constraints on the morphosyntactic

properties of the tokens being reconstructed. x The morphological phenomenon that some words or word classes show instances of systematic homonymy is called syncretism. In particular, homonymy can occur due to neutralization and unaffectedness of words.

Morphological productivity OR **Competence versus performance duality by noam chomsky in the context of morphological productivity.** ‹

x In a natural language as a system (langue), structural devices like recursion, iteration, or compounding allow to produce an infinite set of concrete linguistic utterances. x This general potential holds for morphological processes as well and is called morphological productivity. x In a perspective natural language can be seen as a collection of utterances (parole) pronounced or written (performance). Hence for the linguistic corpora, parole and performance data set is practical. x Such corpora are a finite collection of linguistic data that are studied with empirical methods. It can be used for comparison when linguistic models are developed.

 "80/20 rule," of linguistic word corpus:

x Linguistic corpora are a finite collection of linguistic data that are studied with empirical methods. x The set of word forms found in the corpus of a language is referred as its vocabulary. x The members of this set are word types, whereas every original instance of a word form is a word token. x The distribution these words or other elements of language follows the "80/20 rule," also known as the law of the vital few. x It says that most of the word tokens in a given corpus can be identified with just a couple of word types in its vocabulary, and words from the rest of the vocabulary occur much less or rarely in the corpus. x New, unexpected words will always appear in the linguistic data only when it is expanded or enlarged.

Creativity and the issue of unknown words meet to enhance the morphological productivity in a natural language. OR Unexpected words will always appear in the linguistic data only when it is expanded or enlarged. ‹

The word googol is a dictionary word in English. It means something that is a made-up word denoting the number "one followed by one hundred zeros,". x The name of the company Google is an inadvertent misspelling thereof. However, both of these words successfully entered the lexicon of English where morphological productivity started working. x Today we understand English verb to google and nouns like googling or even googlish or googleology.

x This new word google is adopted by other languages, too. This has triggered their own

morphological processes. x In Czech, one says googlovat, googlit 'to google' or vygooglovat, vygooglit 'to google out', googlování 'googling', and so on. x In Arabic, the names are transcribed as ÷nj÷njl 'googol' and ÷nj÷il 'Google'. x Thus we can observe that unexpected words in a language will always appear in the linguistic data only when it is expanded or enlarged.

1.3 Morphological Models: Motivation of using domain-specific languages. ‹

x A Domain Specific Language (DSL) is a specialized programming language that is used for a single purpose. x Various domain-specific languages have been created for achieving intuitive and minimal programming effort. x Pragmatically, a DSL may be specialized to a particular problem domain, a particular problem representation technique, a particular solution technique, or other aspects of a domain. x These special-purpose languages usually introduce idiosyncratic notations of programs and are interpreted using some restricted model of computation. x The motivation for this approach lies in the fact that, historically, computational resources were too limited compared to the requirements and complexity of the tasks being solved. x Other motivations are theoretical given that finding a simple, accurate and yet generalizing model for the practical use in the specific domain. x The design objective of DSL is to get be pure, intuitive, adequate, complete, reusable and elegant language. x Examples of such domain-specific programming languages are HTML, SQL, AWK, GDL, etc.

Dictionary lookup is considered as one of the effective Morphological model

x Morphological model needs a system in which analysing a word form is reduced kept in sync with more sophisticated models of the language. Dictionaries, Databases and Lists are some of such forms. x A dictionary is understood as a data structure that directly enables obtaining some precomputed results i.e. word analyses. x The data structure can be optimized for efficient lookup, and the results can be shared.

x Lookup operations with dictionaries are relatively simple and usually quick. Dictionaries can be implemented, for instance, as lists, binary search trees, tries, hash tables, etc. x Hence dictionary lookup is considered as one of the effective Morphological models.

Drawbacks of enumerative Morphological model:

x Enumerative list is a set of associations between word forms and their desired descriptions. x It is declared by plain enumeration. Hence the coverage of the model is finite and the generative potential of the language is not exploited. x Development, lookup and verification of the association list is tedious, liable to errors, inefficient and

inaccurate unless the data are retrieved automatically from large and reliable linguistic resources. x Despite all that, an enumerative model is often sufficient for the given purpose, deals easily with exceptions, and can implement even complex morphology.

Finite-state morphological:

x Finite-state morphological models are the morphological models in which the specifications written by human programmers are directly compiled into finite-state transducers. x The finite state morphological models can be used for multiple natural languages. x The two popular online tools supporting this approach are XFST (Xerox Finite-State Tool) and Lex Tools.

Finite state transducers. OR FST can translate the infinite regular.

x Finite-state transducers are computational devices extending the power of finite-state automata. x They consist of a finite set of nodes connected by directed edges labeled with pairs of input and output symbols. x In such a network or graph, nodes are also called states, while edges are called arcs. x Traversing the network from the set of initial states to the set of final states along the arcs is equivalent to reading the sequences of encountered input symbols and writing the sequences of corresponding output symbols. x The set of possible sequences accepted by the transducer defines the input language; the set of possible sequences emitted by the transducer defines the output language. x For example, a finite-state transducer could translate the infinite regular language consisting of the Sanskrit words pita, prapita, praprapita,... to the matching words in the infinite regular English language words defined as father, grand-father, great-grand-father. x In finite-state transducers it is possible to invert the domain and the range of a relation, that is, exchange the input and the output. x In finite-state computational morphology, it is common to refer to the input word forms as surface strings and to the output descriptions as lexical strings.

PART – II

Structure of document:

x In human language, words and sentences do not appear randomly but usually have a structure. x For example, combinations of words form sentences - meaningful grammatical units, such as statements, requests, and commands. x Likewise, in written text, sentences form paragraphs - self-contained units of discourse about a particular point or idea.

Importance of document structure in human language:

x In human language or natural language, words and sentences usually have a structure.

This can be combinations of words form sentences - meaningful grammatical units, such as statements, requests, and commands. x Similarly, in written text, paragraphs are the self-contained units about a point or an idea, which is expressed in the form of group of sentences. Following are the some of the reasons why document structure is important in human languages and therefore for natural language processing. x When the structure of documents is extracted, it makes the further processing of text easy in NLP. The NLP tasks that depends on the document structure are, parsing, machine translation and semantic role labelling in sentences. x To improve the reliability of Automatic Speech Recognition (ASR) and human readability, it is important to identify the sentence boundary annotation. Document structure helps in this process. x Document structure helps in breaking apart the input text or speech into topically coherent blocks that provides better organization and indexing of the data. x Thus, in most speech and language processing applications extracting the structure of textual and audio documents is a meaningful and necessary pre-step.

Sentence boundary detection:

x Sentence boundary detection is the problem in natural language processing of deciding where sentences begin and end.

x Sentence detection is an important task, which should be performed at the beginning of a text processing pipeline. x Sentence boundary detection (also called sentence segmentation) deals with automatically segmenting a sequence of word tokens into sentence units. x Natural language processing tools often require their input to be divided into sentences; however, sentence boundary identification can be challenging due to the potential ambiguity of punctuation marks. x In written text in English and some other languages, the beginning of a sentence is usually marked with an uppercase letter, and the end of a sentence is explicitly marked with a period (.), a question mark (?), an exclamation mark or another type of punctuation. x However, in addition to their role as sentence boundary markers, capitalized initial letters are used to distinguish proper nouns, periods are used in abbreviations and numbers and other punctuation marks are used inside proper names. x A character-wise analysis of text allows for a distinction between period characters that are enclosed between two alphanumeric characters, and period characters that are followed by at least one, non-alphabetic character, such as a further punctuation sign, a space, tab or new line. x There are various challenges associated with SBD, for written as well as spoken text and code switching.

Challenges of sentence boundary detection in written text:

x Ambiguous abbreviations and capitalizations are the most common problems of sentence segmentation in written text. x Quoted sentences are more complex and problematic. The primary reason for this is the speaker may have uttered multiple sentences and sentence boundaries inside the quotes are also marked with punctuation marks. x As a result of this an automatic method of sentence boundary detection may result in cutting some sentences incorrectly. In case if the preceding sentence is spoken instead of written, prosodic cues usually mark structure. x "Spontaneously" written texts, such as Short Message Service (SMS) texts or Instant Messaging (IM) texts, tend to be nongrammatical and have poorly used or missing punctuation, which makes sentence segmentation even more challenging. x The automatic systems, such as optical character recognition (OCR) or ASR, aims to translate images of handwritten, typewritten, or printed text or spoken utterances into machine-editable tex. x When the sentences comes from such automatic system, the finding of sentence boundaries must deal with the errors of these systems as well. x For example, OCR system easily confuses periods and commas and can result in meaningless sentences. ASR transcripts typically lack punctuation marks and are usually mono-case.

Challenges of sentence boundary detection in spoken/ conversational text.

x For conversational speech or text or multiparty meetings with ungrammatical sentences and disfluencies, in most cases it is not clear where the boundaries are.

The problem may be redefined for the conversational domain as the task of dialog act segmentation. This is because dialog acts are better defined for conversational speech using a number of mark-up standards such as Dialog Act Mark-up in Several Layers (DAMSL). x For example, the sentence I think so but you should also ask him may be a grammatical sentence as a whole, but for DAMSL and MRDA standards, there are two dialog act tags, one affirmation and one suggestion. Such a modification may be needed for conversation analysis, such as speaker role detection or sentiment analysis. This task can be seen as a semantic boundary detection task instead of syntactic.

Code switching? Why it is considered as a problem in sentence boundary detection:

x Code switching - that is, the use of words, phrases, or sentences from multiple languages by multilingual speakers - is another problem that can affect the characteristics of sentences. For example, when switching to a different language, the writer can either keep the punctuation rules from the first language or resort to the code of the second language (e.g., Spanish uses the inverted question mark to precede questions). x Code switching also affects technical texts for which the meanings of punctuation signs can be redefined, as in Uniform Resource Locators (URLs), programming languages, and

mathematics. We must detect and parse those specific constructs in order to process technical texts adequately. x Conventional rule-based sentence segmentation systems in well-formed texts rely on patterns to identify potential ends of sentences and lists of abbreviations for disambiguating them. x Although rules cover most of these cases, they do not address unknown abbreviations, abbreviations at the ends of sentences, or typos in the input text. x Furthermore, such rules are not robust to text that is not well formed, such as forums, chats, and blogs, or to spoken input that completely lacks typographic cues. Moreover, each language requires a specific set of rules. x Hence code switching is considered as a problem in sentence boundary detection.

Sentence segmentation as a classification problem is more effective than a rule based problem ‹

x Conventional rule-based sentence segmentation systems in well-formed texts rely on patterns to identify potential ends of sentences and lists of abbreviations for disambiguating them. x Sentence segmentation in text usually uses the punctuation marks as delimiters and aims to categorize them as sentence ending/beginning or not. On the other hand, for speech input, all word boundaries are usually considered as candidate sentence boundaries. x Although rules cover most of these cases, they do not address unknown abbreviations, abbreviations at the ends of sentences, or typos in the input text. x Furthermore, such rules are not robust to text that is not well formed, such as forums, chats, and blogs, or to spoken input that completely lacks typographic cues. Moreover, each language requires a specific set of rules. x To improve on such a rule-based approach, sentence segmentation is stated as a classification problem. Given training data where all sentence boundaries are marked, we can train a classifier to recognize them.

Boundary segmentation:

x Topic segmentation (sometimes called discourse or text segmentation) is the task of automatically dividing a stream of text or speech into topically homogeneous blocks. x That is, given a sequence of (written or spoken) words, the aim of topic segmentation is to find the boundaries where topics. x Topic segmentation is an important task for various language-understanding applications, such as information extraction and retrieval and text summarization. x In information retrieval, if long documents can be segmented into shorter, topically coherent segments, then only the segment that is about the user's query could be retrieved. x For multiparty meetings, the task of topic segmentation is inspired by discourse analysis. x For official and well-structured meetings, the topics are segmented according to the agenda items, whereas for more casual conversational-style meetings, the boundaries are less clear. x For conversational speech, the topic boundaries may not be absolute. Hence they are more complex. x In text, topic boundaries are

usually marked with distinct segmentation cues, such as headlines and paragraph breaks. These cues are absent in speech. However, speech provides other cues, such as pause duration and speaker changes. x Topic segmentation is a nontrivial problem without a very high human agreement because of many natural language-related issues and hence requires a good definition of topic categories and their granularities.

1.5 Methods

Discuss Sentence/ Topic segmentation as a boundary classification problem:

x Sentence segmentation and topic segmentation have mainly been considered as a boundary classification problem. x For given a boundary candidate (between two-word tokens for sentence segmentation and between two sentences for topic segmentation), the goal is to predict whether or not the candidate is an actual boundary (sentence or topic boundary). x Formally, let x ℵ X be the vector of features (the observation) associated with a candidate and y ℵ Y be the label predicted for that candidate. The label y can be b for boundary and b– for nonboundary. x This results in a classification problem: given a set of training examples {x, y}train, find a function that will assign the most accurate possible label y of unseen examples x unseen. x Alternatively to the binary classification problem, it is possible to model boundary types using finer-grained categories.

x Gillick suggested that sentence segmentation in text be framed as a three-class problem: sentence boundary with an abbreviation b^a, without an abbreviation b^a – , and abbreviation not at a boundary b^a – . x Similarly, in spoken language, a three-way classification can be made between non-boundaries b– , statement b s, and question boundaries b^q.

Method of classification in sentence or topic segmentation:

x For sentence or topic segmentation, the problem is defined as finding the most probable sentence or topic boundaries. x The natural unit of sentence segmentation is words and of topic segmentation is sentences, with assumption that assume topics typically do not change in the middle of a sentence. x The words or sentences are then grouped into contiguous stretches belonging to one sentence or topic - that is, the word or sentence boundaries are classified into sentence or topic boundaries and non-boundaries. x The classification can be done at each potential boundary i (local modelling); then, the aim is to estimate the most probable boundary type, y^š i, for each candidate example, xi:

$$\hat{y}_i = \underset{y_i \text{ in } y}{\operatorname{argmax}} P\left(y_i \mid x_i\right)$$

Here, the ^ is used to denote estimated categories, and a variable without a š is used to show possible categories. x In local modelling, features can be extracted from the surrounding example context of the candidate boundary to model such dependencies. It is

also possible to see the candidate boundaries as a sequence and search for the sequence of boundary types, $Y^{\wedge} = y1,\quad y^{\wedge}n$, that have the maximum probability given the candidate examples, $X = x1,.\ ,xn$

Categorization of methods according to the type of the machine learning algorithm. **Generative and discriminative categarization methods:**

Generative sequence model

a) It estimate the joint distribution of the observations, $P(X, Y)$ (e.g., words, punctuation) and the labels (sentence boundary, topic boundary).

b) It requires specific assumptions (such as backoff to account for unseen events) and have good generalization properties.

Discriminative sequence model a) It focus on features that characterize the differences between the labeling of the examples. b) Such methods (as described in the following sections) can be used for sentence and topic segmentation in both written and spoken language, with one difference.

c) In text, the category of all boundaries that do not include a potential end-of-sentence delimiter (period, question mark, exclamation mark) is preset to nonsentence or nontopic,

d) A category is estimated for only those word boundaries that include a delimiter, whereas in speech, all boundaries between consecutive tokens are usually considered.

Generative sequence classification methods for sentence and topic segmentation:

x The most commonly used generative sequence classification method for topic and sentence segmentation is the hidden Markov model (HMM). x The probability is written as the following, using the Bayes rule:

$$\hat{Y} = \operatorname*{argmax}_{Y} P(Y \mid X) = \operatorname*{argmax}_{Y} \frac{P(X \mid Y)\,P(Y)}{P(X)} = \operatorname*{argmax}_{Y} P(X \mid Y)\,P(Y)$$

$P(X)$ in the denominator is dropped because it is fixed for different Y and hence does not change the argument of max. x The bigram case is modelled by a fully connected m-state Markov model, where m is the number of boundary categories. x The states emit words (sentences or paragraphs) for sentence (topic) segmentation, and the state sequence that most likely generated the word (sentence) sequence is estimated. x State transition probabilities, $P(yi \mid yi\text{í}1)$, and state observation likelihoods, $P(xi \mid yi)$, are estimated using the training data. x The most probable boundary sequence is obtained by dynamic programming. x Below is the conceptual hidden Markov model for segmentation with two states: one for segment boundaries, one for others.

x The bigram case can be extended to higherorder n-grams at the cost of an increased complexity. x For topic segmentation, typically instead of using two states, n states are

used, where n is the number of topics. However, it is not possible in HMM to use any information beyond words, such as POS tags of the words or prosodic cues, for speech segmentation. x Two simple extensions have been proposed: Shriberg et al suggested using explicit states to emit the boundary tokens, hence incorporating nonlexical information via combination with other models. x For topic segmentation, Tur et al. used the same idea and modeled topic-start and topic-final sections explicitly, which helped greatly for broadcast news topic segmentation. The second extension is inspired from factored language models which capture not only words but also morphological, syntactic, and other information. Guz et al proposed using factored HELM (fHELM) for sentence segmentation using POS tags in addition to words.

Discriminative local classification methods: ‹

x A number of discriminative classification approaches, such as support vector machines, boosting, maximum entropy, and regression, are based on very different machine learning algorithms. x While discriminative approaches have been shown to outperform generative methods in many speech and language processing tasks, training typically requires iterative optimization. x In discriminative local classification, each boundary is processed separately with local and contextual features. x No global (i.e., sentence or document wide) optimization is performed, unlike in sequence classification. x For sentence segmentation, supervised learning methods have primarily been applied to newspaper articles. x Many classifiers have been tried for the task: regression trees, neural networks, a C4.5 classification tree, maximum entropy classifiers, support vector machines (SVMs), and naive Bayes classifiers. x Mikheev treated the sentence segmentation problem as a subtask for POS tagging by assigning a tag to punctuation similar to other tokens. For tagging he employed a combination of HMM and maximum entropy approaches.

TextTiling method for topic segmentation: OR **Block comparison and vocabulary introduction methods for topic segmentation.** ‹

x The popular TextTiling method of hearst for topic segmentation uses a lexical cohesion metric in a word vector space as an indicator of topic similarity. x TextTiling can be seen as a local classification method with a single feature of similarity. x Below Fig. Q.35.1 depicts a typical graph of similarity with respect to consecutive segmentation units. The document is chopped when the similarity is below some threshold.

Fig. Q.35.1 : TextTiling example
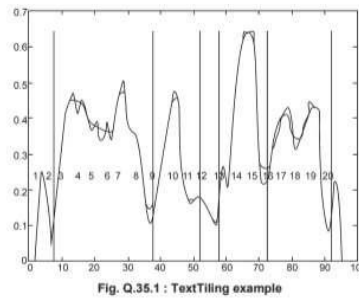
x Originally, two methods for computing the similarity scores were proposed for TextTiling: x Block comparison - a. It compares adjacent blocks of text to see how similar they are according to how many words the adjacent blocks have in common. b. The block size can be variable, not necessarily looking only at the consecutive blocks but instead at a window. c. Given two blocks, b1 and b2, each having k tokens (sentences or paragraphs), the similarity (or topical cohesion) score is computed by the formula given below.

$$\frac{\sum t\, \omega_t\, b_1 \cdot \omega_t\, b_2}{\sqrt{\sum t\, \omega_t^2\, b_1\, \sum t\, \omega_t^2\, b_2}}$$

where ω_t, b is the weight assigned to term t in block b. The weights can be binary or may be computed using other information retrieval-based metrics such as term frequency.

Vocabulary introduction - a. The vocabulary introduction method, assigns a score to a token-sequence gap on the basis of how many new words are seen in the interval in which it is the midpoint. b. Similar to the block comparison formulation, given two consecutive blocks, b1 and b2, of equal number of words, w, the topical cohesion score is computed with the following formula: Where NumNewTerms(b) returns the number of terms in block b, seen for the first time in text.

$$\frac{\text{NumNewTerms}(b_1) + \text{NumNewTerms}(b_2)}{2 \times w}$$

c. This method is extended to exploit latent semantic analysis. Instead of simply looking at all words, researchers worked on the transformed lexical space, which has led to improved results because this approach also captures semantic similarities implicitly.

Discriminative sequence classification methods:

x In segmentation tasks, the sentence or topic decision for a given example (word, sentence, paragraph) highly depends on the decision for the examples in its vicinity. x Discriminative sequence classification methods are in general extensions of local discriminative models with additional decoding stages that find the best assignment of labels by looking at neighbouring decisions to label an example. x Conditional Random Fields (CRFs) are an extension of maximum entropy, SVM struct is an extension of SVM to handle structured outputs, and maximum margin. Markov networks (M3N) are extensions of HMMs. x The Margin Infused Relaxed Algorithm (MIRA) is an online learning approach that requires loading of one sequence at a time during training. x CRFs have been successful for many sequence labelling tasks,

including sentence segmentation in speech. x CRFs are a class of log-linear models for labelling structures. CRFs are trained by finding the 3 parameters that maximize the likelihood of the training data, usually with a regularization term to avoid overfitting.

Gradient, conjugate gradient, or online methods are used for training. x Dynamic programming (Viterbi decoding) is used to find the most probable assignment of labels at test time or to compute the $Z(\cdot)$ function.

Hybrid approaches for word classification: ‹

x Nonsequential discriminative classification algorithms typically ignore the context, which is critical for the segmentation task. x While we may add context as a feature or simply use CRFs, which inherently consider context, these approaches are suboptimal when dealing with real-valued features, such as pause duration or pitch range. Most earlier studies simply tackled this problem by binning the feature space either manually or automatically x An alternative is to use a hybrid classification approach, as suggested by Shriberget al.. x The main idea is to use the posterior probabilities, Pc (yi |xi), for each boundary candidate, obtained from the other classifiers, such as boosting or CRF, by simply converting them to state observation likelihoods by dividing to their priors following the well-known Bayes rule as follows

$$\underset{y_i}{\arg\max} = \frac{P_c(y_i \mid x_i)}{P(y_i)} = \underset{y_i}{\arg\max} P(x_i \mid y_i)$$

Applying the Viterbi algorithm to the HMM then returns the most likely segmentation. To handle dynamic ranges of state transition probabilities and observation likelihoods, a weighting scheme as is usually described in the literature can be applied. x Zimmerman et al. compared various discriminative local classification methods, namely boosting, maximum entropy, and decision trees, along with their hybrid versions for sentence segmentation of multilingual speech. He concluded that hybrid approaches are always superior.

Extensions for global modeling for sentence segmentation

x Most approaches to sentence segmentation have focused on recognizing boundaries rather than sentences in themselves. x This has occurred because of the quadratic number of sentence hypotheses that must be assessed in comparison to the number of boundaries. x To tackle this problem, input is segmented according to likely sentence boundaries established by a local model. Later it is trained as a re-ranker on the n-best lists. x This approach allows leveraging of sentence-level features such as scores from a syntactic parser or global prosodic features. x Favre et al. proposed to extend this concept to a pruned sentence lattice, which allows combining local scores with sentence-level scores in a more efficient manner.

1.6 Complexity of Approaches

Complexity of sentence/topic segmentation is evaluated. ‹

x Sentence/topic segmentation approaches can be rated in terms of complexity (time and memory) of their training and prediction algorithms and in terms of their performance on real-world datasets. Some may also require specific pre-processing, such as converting or normalizing continuous features to discrete features.

x Discriminative approach

a) In terms of complexity, training of discriminative approaches is more complex than training of generative ones because they require multiple passes over the training data to adjust for their feature weights.

x Generative models

b) Generative models such as HELMs can handle multiple orders of magnitude larger training sets and benefit, for instance, from decades of news wire transcripts. But they do

not cope well with unseen events.

x Discriminative classifiers

c) They allow for a wider variety of features and perform better on smaller training sets.

d) Predicting with discriminative classifiers is also slower, even though the models are relatively simple (linear or log-linear), because it is dominated by the cost of extracting more features.

x Sequence approaches e) Compared to local approaches, sequence approaches bring the additional complexity of decoding: finding the best sequence of decisions requires evaluating all possible sequences of decisions. f) Fortunately, conditional independence assumptions allow the use of dynamic programming to trade time for memory and decode in polynomial time. g) This complexity is then exponential in the order of the model (number of boundary candidates processed together) and the number of classes (number of boundary states). x Discriminative sequence classifiers, h) For example CRFs, also need to repeatedly perform inference on the training data, which might become expensive.

1.7 Performance of sentence segmentation approaches:

Short note on: a) Sentence segmentation in speech b) Sentence segmentation in text

c) Sentence segmentation in speech ‹

R a) Sentence segmentation in speech x For sentence segmentation in speech, performance is usually evaluated using - 1) The error rate (ratio of number of errors to the number of examples) 2) F1-measure (the harmonic mean of recall and precision) OR Write a short note on: a) Sentence segmentation in speech b) Sentence segmentation in text c) Sentence segmentation in speech ‹ Ans.: R a) Sentence segmentation in speech x For sentence segmentation in speech, performance is usually evaluated using - 1) The error rate (ratio of number of errors to the number of examples) 2) F1-measure (the harmonic mean of recall and precision)

where 1) Recall is defined as the ratio of the number of correctly returned sentence boundaries to the number of sentence boundaries in the reference annotations.

2) Precision is the ratio of the number of correctly returned sentence boundaries to the number of all automatically estimated sentence boundaries), and the National Institute of Standards and Technology (NIST) error rate (number of candidates wrongly labeled divided by the number of actual boundaries).

R b) Sentence segmentation in text x For sentence segmentation in text, researchers have reported error rate results on a subset of the Wall Street Journal Corpus of about 27,000 sentences. x For instance, Mikheev reports that his rule-based system performs at an error rate of 1.41%. x The addition of an abbreviation list to this system lowers its error rate to 0.45% and combining it with a supervised classifier using POS tag features leads to an error rate of 0.31%. x Without requiring handcrafted rules or an abbreviation list, Gillick's SVM-based system obtains even fewer errors, at 0.25%. x Even though the error rates presented seem low, sentence segmentation is one of the first processing steps for any NLP task, and each error impacts subsequent steps, especially if the resulting sentences are presented to the user as for example, in extractive summarization. R c) Sentence segmentation in speech x For sentence segmentation in speech, Doss et al. report on the Mandarin TDT4 Multilingual Broadcast News Speech Corpus, an F1-measure using the same set of features is as of

o 69.1% for a MaxEnt classifier
o 72.6% with Adaboost
o 72.7% with SVMs x A combination of the three classifiers using logistic regression is also proposed.

not cope well with unseen events.

x Discriminative classifiers

e) They allow for a wider variety of features and perform better on smaller training sets.

f) Predicting with discriminative classifiers is also slower, even though the models are relatively simple (linear or log-linear), because it is dominated by the cost of extracting more features.

x Sequence approaches e) Compared to local approaches, sequence approaches bring the additional complexity of decoding: finding the best sequence of decisions requires evaluating all possible sequences of decisions. f) Fortunately, conditional independence assumptions allow the use of dynamic programming to trade time for memory and decode in polynomial time. g) This complexity is then exponential in the order of the model (number of boundary candidates processed together) and the number of classes (number of boundary states). x Discriminative sequence classifiers, h) For example CRFs, also need to repeatedly perform inference on the training data, which might become expensive.

1.8 Performance of sentence segmentation approaches:

Short note on: a) Sentence segmentation in speech b) Sentence segmentation in text

c) Sentence segmentation in speech ‹

R a) Sentence segmentation in speech x For sentence segmentation in speech, performance is usually evaluated using - 1) The error rate (ratio of number of errors to the number of examples) 2) F1-measure (the harmonic mean of recall and precision) OR Write a short note on: a) Sentence segmentation in speech b) Sentence segmentation in text c) Sentence segmentation in speech ‹ Ans.: R a) Sentence segmentation in speech x For sentence segmentation in speech, performance is usually evaluated using - 1) The error rate (ratio of number of errors to the number of examples) 2) F1-measure (the harmonic mean of recall and precision)

where 1) Recall is defined as the ratio of the number of correctly returned sentence boundaries to the number of sentence boundaries in the reference annotations.

3) Precision is the ratio of the number of correctly returned sentence boundaries to the number of all automatically estimated sentence boundaries), and the National Institute of Standards and Technology (NIST) error rate (number of candidates wrongly labeled divided by the number of actual boundaries).

R b) Sentence segmentation in text x For sentence segmentation in text, researchers have reported error rate results on a subset of the Wall Street Journal Corpus of about 27,000 sentences. x For instance, Mikheev reports that his rule-based system performs at an error rate of 1.41%. x The addition of an abbreviation list to this system lowers its error rate to 0.45% and combining it with a supervised classifier using POS tag features leads to an error rate of 0.31%. x Without requiring handcrafted rules or an abbreviation list, Gillick's SVM-based system obtains even fewer errors, at 0.25%. x Even though the error rates presented seem low, sentence segmentation is one of the first processing steps for any NLP task, and each error impacts subsequent steps, especially if the resulting sentences are presented to the user as for example, in extractive summarization. R c) Sentence segmentation in speech x For sentence segmentation in speech, Doss et al. report on the Mandarin TDT4 Multilingual Broadcast News Speech Corpus, an F1-measure using the same set of features is as of

o 69.1% for a MaxEnt classifier

o 72.6% with Adaboost

o 72.7% with SVMs x A combination of the three classifiers using logistic regression is also proposed.

UNIT II

Syntax Analysis: Parsing Natural Language,

Treebanks: A Data-Driven Approach to Syntax, Representation of Syntactic Structure, Parsing Algorithms, Models for Ambiguity Resolution in Parsing, Multilingual Issues

3.1 : Parsing Natural Language
Problems by using context free grammar for syntactic analysis of natural language.

‹

x Parsing of a natural language is a process of determining syntactic structure of the text by analyzing the words based on underlying grammer.

x It identifies the information which is not explicity given in the input sentence.

x For doing this task it requires some additional information i.e. grammar of a language.

x Context Free Grammar (CFG) is used to write the rules of syntax for ex.:

S → NP VP

NP → 'Atul' | 'pockets' | D N | N P PP

VP → V NP | VP PP

V → ' bought'

D → a

N → ' shirt'

PP → 'PNP'

P → 'With'

Where, V = Verb, NP = Noun phrase, VP = Verb phrase PP = Prepositional phrases

x Typically the rules are stated in the form X o w; Where X is a part of speech for word w which is a generated terminal symbol.

For ex.: In rule V o 'saw', V is a part of speech or preterminal symbol which generates verb saw.
x The sentence 'Atul bough a shirt with pocket's can have two possible forms based on rules stated in CFG.

x In the first parse form pockets can be considered as a currency to buy shirt and in second form the sentence can be of meaning Atul is purchasing shirts with pockets.

1ˢᵗ parse	2ⁿᵈ parse
(S (NP Atul)	(S (NP John)
(VP (VP (V bought)	(VP (V bought)
(NP (D a)	(NP (NP (D a)
(N shirt)))	(N shirt))
(PP (P with)	(PP (P with)
(NP pockets))))	(NP pockets)))))
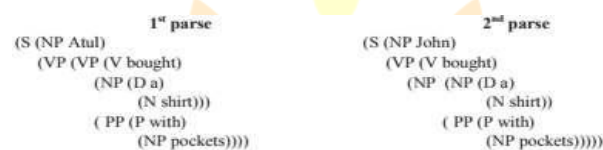
x From the above example we understand that writing all possible syntactic formations in a language is difficult and complex task as we cannot list down a single form. All the possibilities need to explored based on the part of speech tags mentioned for a particular word. x It is also difficult to mention lexical properties of a particular word, which is a first knowledge acquisition problem which is a process of extracting structuring and organizing knowledge. x Second, problem of knowledge acquisition arises from the fact that, it is not only sufficient to know the syntactic rules of a language but it is also required to understand which analysis is most feasible for the sentence in terms of meaning due to ambiguity in syntactic analysis.
x For ex: Consider
CFG N → NN
N → 'natural' | 'language' | 'processing' | 'book'
x If we consider input sentence as natural language processing, two possible ambiguous parses can be generated.

Parse 1	Parse 2
(N (N (N natural)	(N (N natural)
(N language))	(N (N language)
(N processing))	(N processing)))

3.2 : Treebanks:
A Data Driven Approach of Syntax **OR** How to knowledge acquisition problems of context free grammer is addressed by tree bank approach. **OR** Write a note on Treebank approach of syntax analysis. ‹
x Due to ambiguous nature of a language two knowledge acquisition problems arise in syntaclic analysis
1. Due to difficulty in exploration of all the possibilities based on part of speech tags mentioned for a particular word.
2. Due to difficulty in understanding the feasible meaning of sentence due to ambiguity in syntactic analysis.
x To address these problems a data driven approach called as treebank can be adapted.
x Treebank is collection of sentences.
x Each sentence in treebank contains complete syntax analysis.
Human expert provides feasible analysis of the sentence. Annotation guidelines are provided before annotation process.
x Treebank contains annotations of syntactic structure for large set of sentences.
x Supervised machine learning techniques are used to train the parser from the training data extracted from tree banks.
x The first knowledge acquition problem is addressed by providing syntactic analysis

directly instead of grammer.

x The second knowledge acquisition problem is solved as for each sentence in a treebank the most feasible syntactic analysis is provided. x Using supervised machine learning techniques are used to learn scoring function for all possible syntax analysis.

3.3 : Representation of Syntactic Structure
Syntax analysis is done using dependency graphs. ‹

x Dependency graphs connects head of a phrase to its dependents.

x According to definition of Co NLL, in dependency graph nodes are words of input sentence and arcs are binary relations from head to dependent.

x It labelled dependency parsing a label is assigned to each dependency relation between head and dependent word.

x The example dependency graph for Czech sentence from Prague Dependency Treebank is shown in Fig. Q.3.1.

Fig. Q.3.1: An example of a dependency graph syntax analysis for a Czech sentence taken from Prague Dependency Treebank. Each node in the graph is a word, its part of speech, and the not of the word in the sentence, for example [fakulte, N3, 7] is the seventh wood in the sentence with F tag N3, which also tells us that the word has dative case. The node [#, ZSB,0] is the root node of dependency tree. The English equivalent is provided for each node.

Projectivity **in dependency analysis. OR What is projective dependency tree. ‹**
Projectivity is the constraint on syntactic analysis due to effect of linear order of words on dependencies between words. x The example shown in Fig. Q.4.1 shows the english
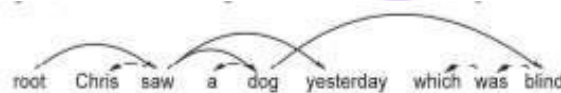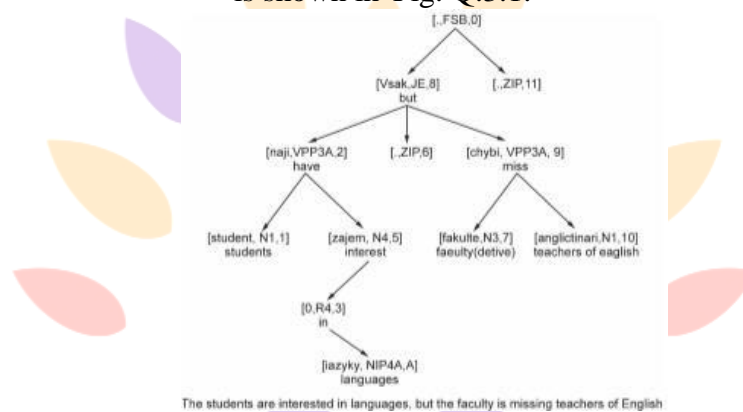

Fig. Q.4.1 : An unlabeled nonprojective dependency tree with a crossing dependency

sentence with the requirements of crossing dependencies.

Syntax analysis using phrase structure tree. ‹
Ans.: x Phrase structure syntax analysis is based on the concept that the sentences can be divided in constituents and larger constituents are formed by merging smaller ones.

x For ex.: Consider the sentence from penn Treebank "Mr. Baker especially sensitive"
The predicate argument structure can be written

directly instead of grammer.

x The second knowledge acquisition problem is solved as for each sentence in a treebank the most feasible syntactic analysis is provided. x Using supervised machine learning techniques are used to learn scoring function for all possible syntax analysis.

3.4 : Representation of Syntactic Structure
Syntax analysis is done using dependency graphs. ‹

x Dependency graphs connects head of a phrase to its dependents.

x According to definition of Co NLL, in dependency graph nodes are words of input sentence and arcs are binary relations from head to dependent.

x It labelled dependency parsing a label is assigned to each dependency relation between head and dependent word.

x The example dependency graph for Czech sentence from Prague Dependency Treebank is shown in Fig. Q.3.1.

Fig. Q.3.1: An example of a dependency graph syntax analysis for a Czech sentence taken from Prague Dependency Treebank. Each node in the graph is a word, its part of speech, and the not of the word in the sentence, for example [fakulte, N3, 7] is the seventh wood in the sentence with F tag N3, which also tells us that the word has dative case. The node [#, ZSB,0] is the root node of dependency tree. The English equivalent is provided for each node.

Projectivity in dependency analysis. OR What is projective dependency tree. ‹

Projectivity is the constraint on syntactic analysis due to effect of linear order of words on dependencies between words. x The example shown in Fig. Q.4.1 shows the english
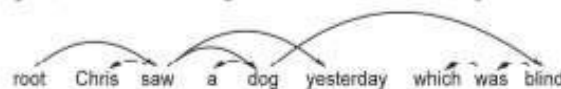

Fig. Q.4.1 : An unlabeled nonprojective dependency tree with a crossing dependency

sentence with the requirements of crossing dependencies.

Syntax analysis using phrase structure tree. ‹

Ans.: x Phrase structure syntax analysis is based on the concept that the sentences can be divided in constituents and larger constituents are formed by merging smaller ones.

x For ex.: Consider the sentence from penn Treebank "Mr. Baker especially sensitive"
The predicate argument structure can be written as,

(S (NP – SBJ (NNP Mr.)

(NNP Baker))

(VP (VBZ seems)

(ADJP – PRD(RB especially)

(JJ sensitive))))

Predicate - argument structure :

Seems ((especially (sensitive)) ((Mr. Banker))

- The subject is marked with - SBJ marker and predicate with - PRD marker.
- The dependency tree can be drawn as,
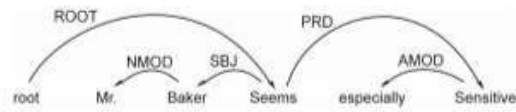
3.5 : Parsing Algorithms OR Explain the working of shift reduce parser. ‹

x In parsing for a given input string we need to do right most derivation of grammar.
x In shift reduce parsing the concept of pushdown automaton (PDA) is used.
x PDA is an automaton that uses stack.

x The shift reduce parser has two steps:

1. Shift step: In this the input stream is advanced by one symbol. The shifted symbol is considered as single node parse tree.
2. Reduce step: It applies completed grammar rule to recent parse trees and combine them together as one tree with new root symbol.
x The working of shift reduce parsing algorithm is shown in Fig. Q.6.1.

Parse Tree	Stack	Input	Action
		a and b or c	Init
a	a	and b or c	Shift a
(N a)	N	and b or c	Reduce N → a
N(a) and	N and	b or c	Shift and
(N a) and b	N and b	or c	Shift b
(N a) and (N b)	N and N	or c	reduce N → b
(N (N a) and (N b))	N	or c	reduce N → a
(N (N a) and (N b)) or	N or	c	shift or
(N (N a) and (N b)) or c	N or c		Shift c
(N (N a) and (N b)) or (N c)	N or N		Reduce N → c
(N (N (N a) and (N b)) or (N c))	N		Reduce N → N or N
(N (N (N a) and (N b)) or (N c))	N		Accept!

Fig. Q.6.1 : The individual steps of the shift-reduced parsing algorithm for the input a and b or c for the grammar G defined at the beginning of this section

- The algorithm is defined as Fig. Q.6.2.

1. Start with an empty stack and the buffer containing the input string.
2. Exit with success if the top of the stack contains the start of the grammar and if the buffer is empty.
3. Choose between the following two steps (if the choice is ambiguous, choose one based on an oracle) :
   - Shift a symbol from the buffer onto the stack.
   - If the top k symbols of the stack are $\alpha_1, \ldots \alpha_k$, which corresponds to the right hand side of a CFG rule $A \rightarrow \alpha_1, \ldots \alpha_k$, then replace the top k symbols with the left-hand side nonterminal A.
4. Exit with failure if no action can be taken in previous step.
5. Else, go to step 2.

Fig. Q.6.2

Hypergraphs and chart parsing. OR Explain CYK algorithm.

x CFG requires the use of database.

x Due to linear parsing technique in CFG in the worst case ran time of algorithm is exponential in the grammar size.

To address this instead of left to right parsing statistical parser which search the space for possible sub trees is used.

x As shown in below example the example grammar can be rewritten in which right hand side contain only two non-terminals.

Example G			New Gc		
N	→	N 'and' N	N	→	N N^
N	→	N 'or' N	N^	→	'and' N
N	→	'a'	'b'	'c'		N	→	N N_v
			N_v	→	N 'or' N
			N	→	'a'	'b'	'c'	

- This can be further made compact by linking input sentence into spans 0 a 1 and 2 b 3 or 4 c 5 etc. i.e. string a is in span 0, 1.

  b or c is in span 2, 5.

- So a new Grammar G_f is written using this concept as below :

New Gf

$$N [0, 5] \rightarrow N [0, 1] N^ [1, 5]$$
$$N [0, 3] \rightarrow N [0, 1] N^ [1, 3]$$
$$N^ [1, 3] \rightarrow \text{'and'} [1, 2] N [2, 3]$$
$$N^ [1, 5] \rightarrow \text{'and'} [1, 2] N [2, 5]$$
$$N [0, 5] \rightarrow N [0, 3] N_v [3, 5]$$
$$N [2, 5] \rightarrow N [2, 3] N_v [3, 5]$$
$$N_v [3, 5] \rightarrow \text{'or'} [3, 4] N [4, 5]$$
$$N [0, 1] \rightarrow \text{'a'} [0, 1]$$
$$N [2, 3] \rightarrow \text{'b'} [2, 3]$$
$$N [4, 5] \rightarrow \text{'c'} [4, 5]$$

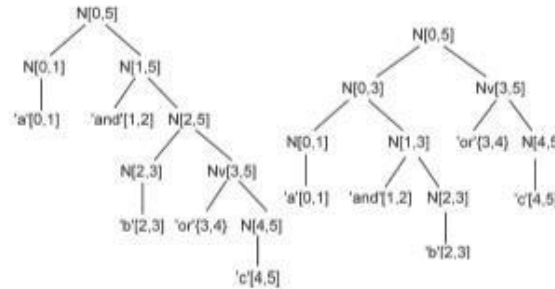- The parse tree that starts from a start nonterminal and spans complete string is shown in Fig. Q.7.1.

Fig. Q.7.1 : Parse trees embedded in the specialized CFG for a particular input string. The nodes with the same label, such as N[0,5], N[0,1] and [1,2], N[2,3] and Nv[3,5] can be merged to form a hypergraph representation of all parses for the input

x To construct this specialized CFG following are steps: x First rules generating lexical items are considered For ex. N [0, 1] o 'a' [0, 1] N [2, 3] o 'b' [2, 3] N [4, 5] o 'c' [4, 5]

x The pseudo code is written as shown in Fig. Q.7.2.

```
for i = 0 ..... n do
    if N → x with score s for any x spanning i, i + 1 exists then
        add specialized rule N [i, i + 1] → x [i, i + 1] with score s
        written as N[i, i + 1] : s
    end if
end for
```

Fig. Q.7.2

- In the next step specialized rules are created recursively. For ex. If Y[i, k] and Z[k, j] are left hand sides of previously created rules then rule rate X → YZ can be converted to

$$X [i, j] → Y [i, k] Z [k, j]$$

- Score S is assigned to each non-terminal span.
- The algorithm is shown in Fig. Q.7.3 and is known as CKY (Cocke, Kasami and younger) algorithm.

```
for j = 2 .... n do
    for i = j − 1 ... 0 do
        for k = i + 1 .... j do
            if Y [i, k] : s₁ and Z[k, j] : s₂ are in the specialized grammar then
                if X → YZ with score s exists in the original grammar then
                    add specialized rule X[i, j] → Y[i, k] Z[k, j] with score s + s₁ + s₂
                    keep only the highest scoring rule : X[i, j] → α
                end if
            end if
        end for
    end for
end for
```

Fig. Q.7.3

**Working of minimum spanning trees and dependency parsing.**
‹

x The Minimum Spanning Tree (MST) corresponds to the optimum branching problem in directed graphs. Which are rooted and does not have cycles.

x The basic prequisite is all the dependency links between the words must have score.

- To understand the working of MST consider the input sentence as

  Devika saw Aastha

- The fully connected graph of this sentence can be drawn as follows,
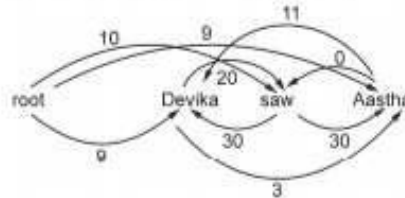
- The scoring function is used to assign the weights to the edges.

- The algorithm starts with finding the incoming edge with highest score.

- We have a cycle in this graph. We can combine the cycle into single node and recalculate edge weight.

- The edge weight from each node to this newly combined node is computed.

- Also record the maximum weight of a node in this combination.

- For example the incoming edge.

  root → [saw → Devika]   is having weight 10 + 30 = 40

  Whereas root → [Devita → saw]   is having weight 9 + 20 = 29

  Aastha → [saw → Devika]   is having weight 0 + 30 = 30

  and Aastha → [Devika → saw]   is having weight 11 + 20 = 31
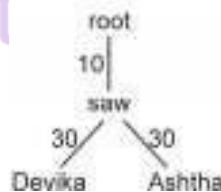
So the graph can be viewed as
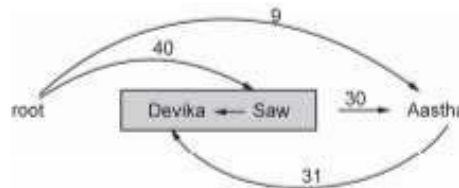
- The MST algorithm is recursively applied to this graph and best incoming edges to each word are found.

- So in the next iteration

  root → Aastha → [Devika → saw]   will have weight 9 + 31 = 40

  and   root → [Devika → saw] → Aastha will have weight 40 + 30 = 70

- So the best nodes chosen are shown in below graph
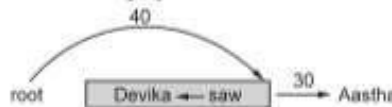
And we get a highest scoring dependency parse as

3.6 : Models for Ambiguity Resolution in Parsing OR How ambiguity resolution in parsing is done with the help of probabilistic context free grammars. ‹

x Consider the example sentence "Atul bought a shirt with pockets" explain a in T.

x Please refer to the CFG and parse trees of this sentence from T1.

x To resolve the ambiguity of such type, one way is to assign the probabilities to the rules in CFG.

x Due to this the CFG is known as Probabilistic Context Free Grammar or PCFG. For example for the rule N $\rightarrow$ D the probability can be defined as P(N $\rightarrow$ D/N) such that rule probability is stated at left hand side.

x Due to this assignment

$$1 = \sum_{\alpha} P(N \rightarrow \alpha)$$

- So for the example sentence the probability distribution can be viewed as

  S $\rightarrow$ NP VP (1.0)

  NP $\rightarrow$ 'Atul' (0.1) | 'pockets' (0.1) | DN (0.3) | NP PP (0.5)

  VP $\rightarrow$ V NP (0.9) | VP PP (0.1)

  V $\rightarrow$ 'bought' (1.0)

  D $\rightarrow$ 'a' (1.0)

  N $\rightarrow$ 'shirt' (1.0)

  PP $\rightarrow$ P NP (1.0)

  P $\rightarrow$ 'with' (1.0)

- From the assigned probabilities we can observe that the decision should be taken from two rules NP $\rightarrow$ NP PP and VP $\rightarrow$ VP PP and as probability of NP $\rightarrow$ NP PP is having higher probability it generates a more feasible sentence.

- Consider the following example of derivation of rule probabilities from treebank. The figure below shows treebank with three trees t_1, t_2 and t_3.
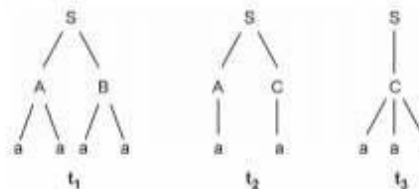
Assume that t_1 occur 10 times, t_2 : 20 times and t_3 : 50 times in treebank.

PCFG for this can be given as
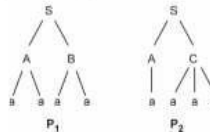
$$\frac{10}{10+20+50} = 0.125 \qquad\qquad S \rightarrow A\,B$$

$$\frac{20}{10+20+50} = 0.25 \qquad\qquad S \rightarrow A\,C$$

$$\frac{50}{10+20+50} = 0.625 \qquad\qquad S \rightarrow C$$

$$\frac{10}{10+20} = 0.334 \qquad\qquad A \rightarrow a\,a$$

$$\frac{20}{10+20} = 0.667 \qquad\qquad A \rightarrow a$$

$$\frac{20}{20+50} = 0.285 \qquad\qquad B \rightarrow a\,a$$

$$\frac{50}{20+50} = 0.714 \qquad\qquad C \rightarrow a\,a\,a$$

- If we consider input a a a a, we can generate two parses.

Where, $\quad P_1 = 0.125 * 0.334 * 0.285 = 0.01189$

$\qquad\qquad P_2 = 0.25 * 0.667 * 0.714 = 0.119$

So P_2 is most feasible for parsing.

Generative models for parsing for ambiguity resolution.

x A parse tree is typically built by sequence of decisions.
x Based on the CFG rules there can be multiple derivations.
x Consider each such derivation as, D = d1, ..., dn x Parser has to choose from these derivations, the most feasible one.
x Let us consider input sentence as x and output parse tree need to be generated as y.
x For each derivation of parse tree the probability can be assigned as

$$P(x, y) = P(d_1, \ldots, d_n) = \prod_{i=1}^{n} (d_i \mid d_1, \ldots, d_{i-1})$$

- In this equation a partial parse tree is built by probability $(d_i \mid d_1, \ldots, d_{i-1})$ which is known as history.
- These histories are grouped into conditional classes by function ϕ as :

$$P(d_1, \ldots, d_n) = \prod_{i=1}^{n} (d_i \mid \phi(d_1, \ldots, d_{i-1}))$$

- History $H_i = d_1, \ldots, d_{i-1}$ for all x, y is represented by finite set of feature functions k

$$\phi_1(H_i), \ldots \phi_k(H_i)$$

$$P(d_1, \ldots, d_n) = \prod_{i=1}^{n} [d_i \mid \phi_1(H_i), \ldots, \phi_k(H_i)]$$

Global linear discriminative model ambiguity resolution parsing.

x Discriminative model in developed by Collins for creating simple framework for describing discriminative approaches, which is also called as global linear model.
x Consider X as set of inputs and Y as output which is sequence of POS tags as parse

- Each input $x \in X$ and $y \in Y$ is mapped to d-dimensional feature vector $\phi(x, y)$. Each dimension is a real number.
- Weigh is assigned to each feature is $\phi(x, y)$ by weight parameter vector $w \in R^d$.

$$\phi(x, y) \cdot w = \text{Score of } (x, y)$$

If the score is higher y is the most feasible output for x.

- Possible outputs y from x are computed from function GEN(x).
- The highest scoring candidate y* from GEN(x) is computed as

$$F(x) = \underset{y \in \text{GEN}(x)}{\text{argmax}} \ P(y \mid x, w)$$

- Conditional random field C (RF) computer conditional probability as

$$\log p(y \mid x, w) = \phi(x, y) \cdot w - \log \sum_{y' \in \text{GEN}(x)} \exp[\phi(x, y') \cdot w]$$

- Global linear model is stated as

$$F(x) = \underset{y \in \text{GEN}(x)}{\text{argmax}} \ \phi(x, y) \cdot w$$

Original perceptron learning algorithm for ambiguity resolution in parsing.
x Perceptron is a single layered neural network.
x It process a example at a time.
x The weight adjustment is done of weight parameter vector.
x This vector is applied to input to generate the related output.
x The features present in the truth are "recognized.
x Consider a training set with in examples.

• As shown in Fig. Q.12.1 Algorithm 12.1 the original perceptron learning Q.12.1 Algorithm works like below.

```
Algorithm 12.1 : The original perceptron learning algorithm
Inputs : Training data { (x₁, y₁), ....... (xₘ, yₘ) }; number of iterations T
Initialization : Set w = 0
Algorithm :
1.   for t = 1, .........., T do
2.   for i = 1, .........., m do
3.   Calculate yᵢ, where y'ᵢ = argmax Φ(xᵢ, y) · w
              y ∈ GEN (x)
4.   if  y'ᵢ ≠ yᵢ then
5.   w = w + Φ(xᵢ, yᵢ) − Φ(xᵢ, y'ᵢ)
6.   end if
7.   end for
8.   end for output : The updated weight parameter vector w
```

Fig. Q.12.1 : Algorithm 12.1

1. Weight parameter w is initialized to 0.
2. Iteration is carried out on m training examples.
3. Set of candidates GEN(x) is generated for each x.
4. The most feasible candidate i.e. the candidate with maximum score according to w is selected.
5. w is updated by increasing weight values of features in truth and decreasing weight value for features appearing in top candidate.
x The problem faced by this algorithm is of overfitting during incremental weight update due to which unseen data is not classified properly.
x The algorithm is not suitable for linearly inseparable training data.

Voted perceptron algorithm for ambiguity resolution in parsing.
x As shown in Fig. Q.13.1 Algorithm 13.1 the voted perceptron algorithm, proposed by freund and Schapire and works as follows:
o Instead of a single weight vector W, the learning process considers all intermediate weight vectors.
o In classification phase-these intermediate vectors are used to vote for the answer.
o Good prediction vector generally service for long time and have larger weight in the vote.
o In training phase count Ci counts the survival of weight parameter vector (wi, Ci) in training.
o If top candidate is not in truth Ci + 1 is initialized 1, to generate an updated weight vector (wi + 1, Ci + 1).
o While this original Ci and weight vector (Wi, Ci) are stored.
o This algorithm is more stable - than original perceptron.

Fig. Q.13.1 : Algorithm 13.1

Averaged perceptron algorithm for ambiguity resolution in parsing. ‹

x The averaged perception algorithm reduces space and time complexities.

x As shown in Fig. Q.14.1 Algorithm 14.1 instead of w-he averaged weight parameter vector J on m training examples is used.

x J can be defined as

$$\gamma = \frac{1}{mT} \sum_{i=1,\ldots,m,\, t=1\ldots T} w^{i,t}$$

- Accumulating parameter vector σ is maintained and updated using w.

```
Algorithm 14.1 : The averaged perceptron learning algorithm
Input : Training Data < (x₁, y₁), ...... (xₘ , yₘ) }; number of iterations T
Initialization : Set w = 0, γ = 0,  σ = 0
Algorithm :
for t = 1, ..........., T do
          for i = 1, ............. , m do
          Calculate yᵢ , where yᵢ'  = argmax Φ(xᵢ , y) · w
                                          y ∈ GEN (x)
          if   yᵢ' ≠ yᵢ then
             w = w + Φ(xᵢ , yᵢ) – Φ(xᵢ , yᵢ')
          end if
                    σ = σ + w
     end for
end for
output : The averaged weight parameter vector γ = σ / (mT)
words, are used as heads and dependent
```

Fig. Q.14.1 : Algorithm 14.1

3.7 : Multilingual Issues OR What are the different issues faced by a parser in terms of tokenization, case and encoding? ‹

x Typically in a grammar a word has a specific definition.

x But the definition of a word may be different for different parsers or treebanks.

For ex. In English language two tokens are separated by space, but in parser for english word today's or there's are considered as two independent tokens i.e. today and 's, there and 's. x Second issue is with uppercase and lowercase words.

x If we simply convert all the uppercase of treebank to the lowercase we may loose some useful information. For ex: In proper nouns like India the first letter will always in a uppercase form but if we lowercase all the letters then in the training data the proper nouns may look like a simple verbs.

x Solution of this problem can be selective lower casing. Also low count token can be replaced by a pattern. For ex.: India appears twice it is replace of by in dates same is applicable to URL's or IP addresses, etc.

x The next issue is with the encoding style of the 'language script'.

x If the script is not in ASCII encoding then the encoding needs to be handled.

x Care should be taken-that data used by parser is converted to the encoding of tree bank and vice versa. x The text data may take different encoding styles (based on the language script) such as UTF-8, GB, BIG5, for Chinese text.

Explain the role of word segmentation in parsing. ‹

x In the languages like Chinese the words are not separated by the identification marks. The written form of many languages, including Chinese, lack marks identifying words.

The written form of many languages, including Chinese, lack marks identifying words. Given the Chinese text,北京大学生比赛, a plausible segmentation would be 北京 (Beijing) 大学生 (university students) / 比赛 (competition) 'competition among university students in Beijing'. However, if 北京大学 is taken to mean Beijing University, the segmentation.

x From the above example we can understand the importance of word segmentation.

x It is process by which the character sequence is divided into blocks so that the generated output is meaningful but consists of separate tokens.

x After identification of all the words in a sentence POS tags are assigned to each word and syntax tree for a sentence can be built.

x This task is easy for the languages like English as all the tokens (words) are separated

by spaces, but quite challenging for the languages like Chinese, Japanese, etc.

x Various researchers have addressed this problem by different approaches.

x In one approach for Chinese text parsing, the parser assigns word boundaries where non terminals of tree specify word boundaries.

x But the immediate context proves to be more useful in detecting word boundaries.

x In the approach stated by Bar-Hillel, Perles and Shamir, the parser consider states in automata as indices. The input is considered as a directed acyclic graph with a single start point and edges labeled with word and height called as word lattice. This word lattice represents multiple segmentation probabilities of the segmented Chinese text. The most feasible ranked segmentation is chosen by the parser to generate most accurate parse.

Explain the role of morphology in syntactic analysis. OR Explain now morphology affects parsing of the languages other than English. ‹

x Basic components of a word are called morphemes.

x Meaning of the word is derived from combination of meanings of morphemes.

x A word is considered to be a combination of different morphemes contributing to the meaning.

x For ex: Consider a Turkish sentence shown in Fig. Q.17.1 '+' symbol shows the morphemes present in a word. So it is feasible to use morphemes as heads and dependents instead of words.

Fig. Q.17.1
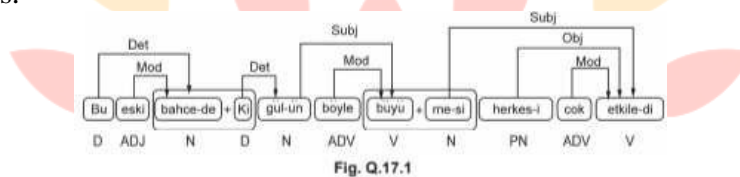
x In the languages like Czech and Russian morphemes are also used to show grammatical case, ganders, etc.

x For Ex.: In Czech language maximum adjectives form all four ganders, seven case markers, all three degrees of comparison and are positive and negative in polarity, resulting in 336 inflected words only for adjectives.

x One solution to resolve the ambiguity in morphology is to assign POS tag for encoding various morphemes.

For ex.: POS V….M-3 ….. Indicates that each word contain morphemes in 10 dimensions and in this example stem is verb with masculine gender in third person.

x This is accomplished by training sub classifies for each component of POS tag.

Semantic Parsing: Introduction, Semantic Interpretation, System Paradigms

Semantic parsing

x Semantic parsing is an important phase of Natural Language Processing (NLP).

x Any document is comprised of set of sentences.

x These sentences are formed by the words arranged according to the rule based grammar as per the constructs in a particular language.

x To extract the meaning out of this syntactically arranged sentences is the objective of semantic parsing.

x Parsing is analysing a text into logical syntactic components and semantics is study of meaning.

x The information pieces are identified and related in semantic parsing.

x The meaningful parts are identified in the text and they are transformed into suitable data structures for higher level task accomplishment.

Types of semantic parsing OR Differentiate between deep semantic parsing and shallow semantic parsing. ‹

x Semantic parsing is about identifying the meaning.

x There are two approaches which can be adapted extracting the meaning out of text.

x Consider domain specific applications like travel reservation, gaming simulation etc.

x In these kind of applications precise and each meaning extraction can be done based on the restricted limits of the particular domain.

x In this approach based on the query or the purpose, output is generated from the meaning representation. This approach is known as deep semantic parsing.

x The second approach is not domain specific and is more generic.

x In this approach the task of meaning representation is divided in small pieces. It is known as shallow semantic parsing.

x These pieces are responsible for capturing small manageable components that represent meaning.

x The related sets of meaning representations are created in this case for example extraction of word sense disambiguation followed by predicate argument structure.

x In case of deep semantic parsing there is little or no scope of reusability as every domain is unique and requires different understanding and representation of the concepts.

x In case of shallow semantic parsing as generic meaningful pieces need to be created, it is very difficult to have general purpose ontology and symbols which are shallow from learning point of view but have high reusability irrespective of the applications.

3.2: Semantic Interpretation

x Semantic interpretation facilitates the joining of different components which define meaning representation of the text.

x This representation, when fed to a computer can be further processed for computational manipulations and search for any application.

x Semantic parsing is the part of semantic interpretation.

Semantic theory.

x The semantic theory was proposed by katz and fodor in 1963.

x It addresses the fact of meaning representation which is lacking in Chomsky's transformational grammer.

x Following points are stated in semantic theory:

1. Ambiguous sentences should be taken care of and explained properly. Consider the word 'bank' in the sentence 'I went to bank'. In this sentence ambiguity exist as there can be two meaning of the work bank first is money bank and second is river bank.

2. Ambiguity resolution of the words depending on context should be done. For example: If the sentence is extended as 'I went to bank to withdraw money' then the theory should be able to disambiguate word 'bank' to extract correct meaning.

3. The theory should be able to identify meaningless but syntactically correct sentences. For example: Consider the sentence 'Colorless green ideas sleep furiously'. This sentence does not carry any meaning but is syntactically correct.

4. Many times same semantic content can be represented by different unrelated syntactic structures such sentences should be handed.

List and explain all the components in semantic interpretation. ‹

The process of semantic interpretation consists of different components to represent a text which can be fed into and processed by computer for undergoing various functionalities like search which are basis of language understanding system.

x Following are the major components in this process

1. Structural ambiguity

x Any sentence is represented by its syntactic structure.

x Syntax and semantics are closely related to each other and we consider that the semantic interpretation is based on underlying syntactic structure.

x So syntactic processing is the first step in semantic interpretation.

2. Word sense

x Many times in any language a same word is used in different meaning depending on the world knowledge.

x For example a word "bank" represents a money bank or it can also posses a meaning river bank.

x Due to inherent intelligence and language vocabulary by humans it is not a difficult for them to understand the meaning of the word expected by speaker or author.

x Consider the examples:

1. Go to bank to withdraw money.

2. Fetch some water from river bank.

x It's a easy task for humans to disambiguate the meaning of word band in above sentences.

x But for a machine this word sense disambiguation is a challenging task and it plays important role in semantic interpretation.

3. Entity and entity resolution

x Any text consists of various entities falling in different categories like person name, locations, quantities, etc.

x Identification of these entities in the major task in semantic interpretation system.

x Named Entity Recognition (NER) is a subtask of information extraction for locating and classifying named entities in the above mentioned categories.

` • For example : [Devika]$_{Person}$ bought 100 shares of [Infosis]$_{organization}$ in [2019]$_{time}$

x Another important task is of co-reference resolution.

x Co-reference occurs when two or more expressions in text refer to same person. For example: Pranjali said she will sing. In this sentence proper noun Pranjali and she refers to same person. x This task is also under information extraction and a major component in semantic interpretation.
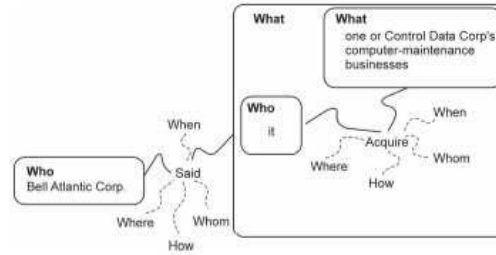
❑ 4. Predicate argument structure

Fig. Q.5.1 : A representation of *who* did *what* to *whom, when, where, why* and *how*

After finishing all the above mentioned tasks of word sense disambiguation, NER and coreference resolution, next task is to identify the participants in the events.

x It is important to identify what is a role of entity in a particular event.

x This is known as resolving the argument structure of predicate in a sentence.

x It is typically identification of who did what to whom, when, where and how as shown in Fig. Q.5.1.

5. Meaning representation

 x This is the last step in semantic interpretation and also called as deep representation.

 x It involves building meaning representation which can be used by algorithms for various applications.

x Research is still going on in the area of general purpose representations.

x Much study has been done in domain specific applications.

x For example: Consider a Geoquery domain, the example query can be represented as which river is the longest? (x1, longest (x, river (x1)))

3.3: **System Paradigms OR What are the different approaches for practical simple mentation of semantic interpretation?** ‹

x Based on the diversity of languages and levels of granularity and generality there can be different approaches to handle semantic representation.

x The main constraint is posed by lack of data availability of the languages.

x Due to these constraints some successful approaches for practical implementation of semantic interpretation fall in three categories:

1. System architectures

1. Knowledge based: These systems are designed using predefined set of rules for finding the solution to a particular problem.

2. Unsupervised: Solution to a particular application is found by minimum human involvement.

3. Supervised: This technique involves model training by application of different machine learning algorithms. Feature functions are created to create features which are used to predict labels to handle unseen data.

4. Semi supervised: Supervised technique involves more human involvement so is expensive. So the data set can be expanded by application of machine generated output directly.

2. Scope

1. Domain dependent: Systems designed for specific domains such as travel reservations or football coaching.

2. Domain independent: It includes generic techniques applicable for multiple domains.

3. Coverage: 1. Shallow: In this approach intermediate representation is generated to be used by machine. 2. Deep: Through this approach representation which is created, directly used by machine.

Availability of the resources for word sense disambiguation. OR What are the different resources for word sense disambiguation in language understanding?

x In case of natural language understanding the availability of the corpus is the crucial factor.

x Uptil now the inadequacy of the tagged sense data is the major issue faced by language understanding systems.

x In case of word sense disambiguation the resources are evolved from machine readable dictionaries to the use of word net in the recent systems.

x In early days Longman Dictionary of Contemporary English (LDOCE) and Roget's thesaurus were the resources used for task of disambiguation.

x In late 1980's WordNet was developed which contributed significantly in this task.

x WordNet is a powerful resource which contains lexical database of word senses with multiple parts of speech of a language.

x It also handles different relationships like hyperngmy, homonymy, metonymy etc. connecting different words.
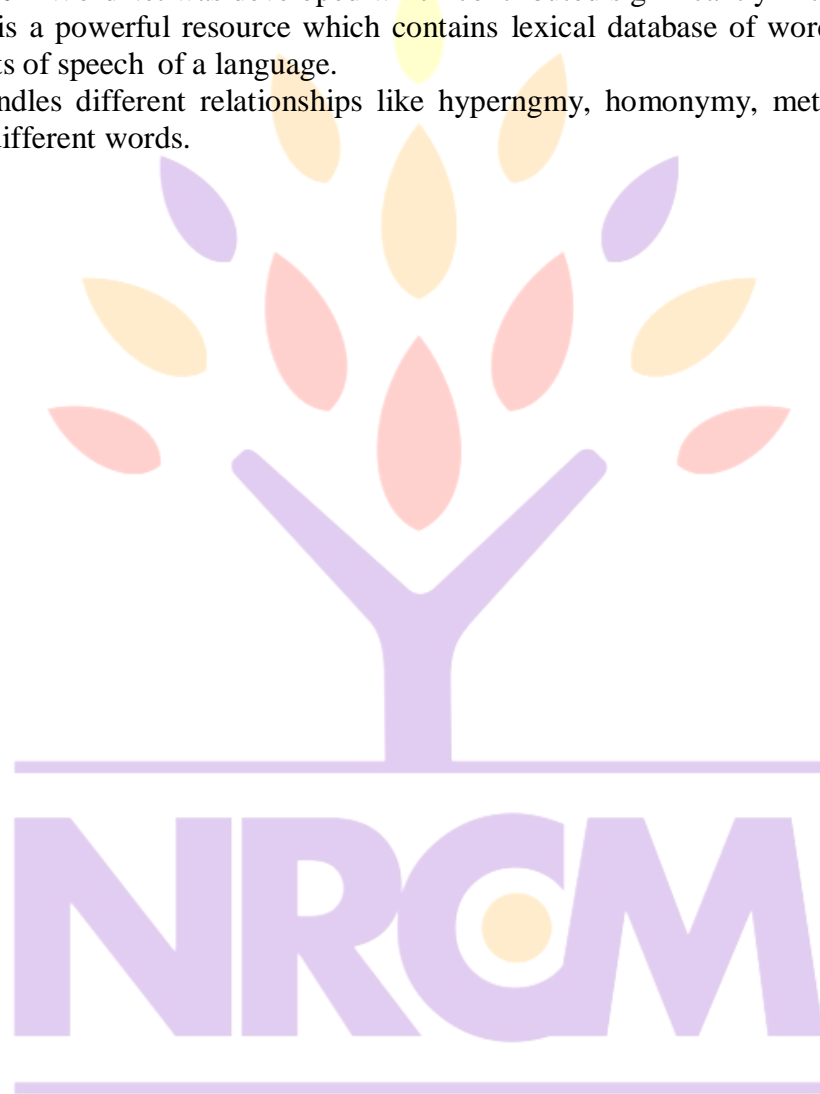
Predicate: Argument Structure, Meaning Representation Systems, Software.

Discourse Processing: Cohesion, Reference Resolution, Discourse Cohesion and Structure

4.1 Predicate Argument Structure OR What are the approaches for converting linguistic insights into features? ‹

x FrameNet and PropBank are two major semantically tagged corpora.

x They have evolved from rule based approaches to data oriented approaches.

x They transform linguistic understanding to features other than rules.

Machine learning techniques can use these features to train the model which can be used to automatically tag the semantic information which is present in these resources.

x FrameNet

o FrameNet is a project housed at the international computer science institute in Berkeley, California which produces an electronic resources based on a theory of meaning called frame semantics.

o Frame semantics is a theory of linguistic meaning developed by Charles J. Fillmore that extends his earlier case grammar. It relates linguistic semantics to encyclopedic knowledge.

o The basic idea is that one cannot understand the meaning of a single word without access to all the essential knowledge that relates to that word.

o For example, one would not be able to understand the word "sell" without knowing anything about the situation of commercial transfer, which also involves, among other things, a seller, a buyer, good, money, the relation between the money and the goods, the relations between the seller and the goods and the money, the relation between the buyer and the goods and the money and so on.

x PropBank

o PropBank is a corpus that is annotated with verbal propositions and their arguments - a "proposition bank".

o Although "ProbBank" refers to a specific corpus produced by Martha Palmer et al., the term PropBank is also coming to be used as a common noun referring to any corpus that have been annotated with proposition and their arguments.

FrameNet. ‹

x FrameNet is a project developed at the international computer science Institute in Berkeley, California by Charles J. Fillmore.

x It is based on the concept of theory of meaning called frame semantics.

x A semantic frame is a conceptual structure describing an event, relation or object and the participants in it.

x For e.g. Consider the example: "Atul sold a house to Anil" describes the same fact as "Anil bought house from Atul". This same basic situation is known as a semantic frame.

x The FrameNet lexical database contains over 1, 200 semantic frames, 13000 lexical units (a pairing of a word with a meaning; polysemous words are represented by several lexical units) and 202,000 example sentences.

x The goal of FrameNet project is to facilitate automatic semantic RoleLabeling.

x It contains semantic annotation of predicates in English and also contain tagged sentences from British National Corpus (BNC).

x The process involves: o Identification of frames invoked by predicates and creation of frame specific roles called as frame elements.

o Identification of predicates which instantiated the frame and label the sentences for those predicates.

x The labeling process involves:

o Identification of frame instantiated and invoked by predicate lemma.

o Identification of semantic arguments for that instance.

o Tagging these arguments with predetermined set of frame elements for that particular frame.

x Combination of predicate lemma and the frame invoked by its instance is known as Lexical Unit (LU).

x For e.g.: Consider The DRIVING frame, for example, specifies a DRIVER (a principal MOVER), a VEHICLE (a particularization of the MEANS element) and potentially CARGO or RIDER as secondary movers. In this frame, and DRIVER initiates and controls the movement of the VEHICLE. For most verbs in this frame, DRIVER or VEHICLE can be realized as subject; VEHICLE, RIDER or CARGO can appear as direct objects; and PATH and VEHICLE can appear as oblique complements.

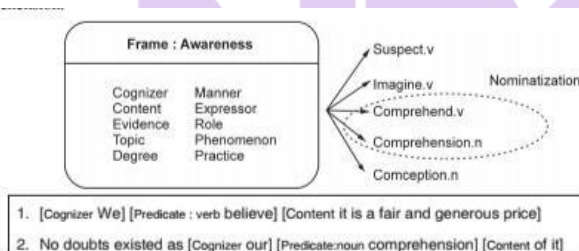x For example in Fig. Q.2.1 Frame is: AWARENESS, verb predicate: Believe, noun predicate: comprehension.

Fig. Q.2.1 : FrameNet example

PropBank.

x PropBank is a corpus of text annotated with information about basic semantic propositions created by Martha Palmer.

x It is created by adding predicate-argument relations to the syntactic trees of Penn Treebank.

x In PropBank the arguments of each predicate are annotated with their semantic roles in relation to the predicate.

x PropBank annotation also requires the choice of a sense id (also known as a 'frameset' or 'roleset' id)for each predicate.

x For each verb in every tree which presents the phrase structure of the sentence, a PropBank instance is created which consists of the sense id of the predicate and its arguments labeled with semantic roles.

x It provides consistent argument labels across different syntactic realizations of the same verb.

For e.g.: [ARG0 Devika]broke [ARG1 the window]

[ARG1 The window] broke

In the above example the arguments of the verbs are labeled as numbered arguments: Arg0, Arg1, Arg2 and so on.

x As shown in Table Q.3.1 the arguments are tagged as core arguments (label type ARGN, where N = 0 to 5) or adjunctive arguments (label type ARGM-X, where X = TMP for temporal, LOC for locative, etc.)

Table Q.3.1 : List of adjunctive arguments in PropBank - ARGMS

Tag	Description	Example
ARGM-LOC	Locative	the museum, in Westborough, Mass
ARGM-TMP	Temporal	now, by next summer
ARGM-MNR	Manner	heavily, clearly, at a rapid rate
ARGM-DIR	Direction	to market, to Bangkok
ARGM-CAU	Cause	In response to the ruling
ARGM-DIS	Discourse	for example, in part, Similarly
ARGM-EXT	Extent	at $38.375, 50 points
ARGM-PRP	Purpose	to pay for the plant
ARGM-NEG	Negation	not, n't
ARGM-MOD	Modal	can, might, should, will
ARGM-REC	Reciprocals	each other
ARGM-PRD	Secondary predication	to become a teacher
ARGM	Bare ARGM	with a police escort
ARGM-ADV	Adverbials	(none of the above)

• For e.g. Core arguments for predicates operate and author are shown in Table Q.3.2.

Table Q.3.2 : Argument labels associated with the predicate operate.01 (sense : work) and for author.01 (sense : to write or construct) in the PropBank corpus

Predicate	Argument	Description
operate.01		
	ARG0	Agent, operator
	ARG1	Thing operated
	ARG2	Explicit patient (thing operated on)
	ARG3	Explicit argument
	ARG4	Explicit instrument
author.01		
	ARG0	Author, agent
	ARG1	Text authored

Semantic role labeling using Semantic Role Labeling (SRL) algorithm. x Gildea and Jurafsky proposed that semantic role labeling is a unsupervised classification problem, in which predicate and arguments of predicate are mapped to a node in a syntax tree for a particular sentence.

x Semantic role labeling consist of three tasks: R Argument identification: Involves identification of all the parse constituents representing valid semantic arguments of a predicate. R Argument classification: Involves assigning appropriate argument labels to identified constituents. R Argument identification and classification: It is a combination of above two tasks i.e. identification of constituents and assigning the label to them.

x The process can be explained through Semantic Role Labeling (SRL) algorithm shown in Fig. Q.4.1. Procedure: SRL (sentence) returns best semantic role labeling Input: Syntactic parse of the sentence

1. Generate a full syntactic parse of the sentence

2. Identify all the predicates

3. For all predicate • sentence do

4. Extract a set of features for each node in the tree relative to the predication

5. Classify each feature vector using the model created in training

6. Select the class of highest scoring classifier

7. Return best semantic role labeling

8. end for

Explain the features stated by Gildea and Jurafsky in Phrase Structure Grammar (PSG) for semantic role labeling problem. ‹

x In semantic role labeling problem apart from frameNet and propBank, certain high quality statistical parsers can be used to generate a phrase structure tree.

x As phrase structure is open for modifications Gildea and Jurafsky used some features to address semantic role labeling problem.

x These features are explained below

1. Path

o Consider the example in Fig. Q.5.1, the path which starts from ARGO. It to predicate operates follows the string.

NP $\rightarrow$ S $\rightarrow$ VP $\rightarrow$ VBZ

OR NP n| S p VP p VBZ

Where n = Upward movement, p = Downward movement in tree

This is known as a Syntactic Path from parse constituent to predicate be classified.
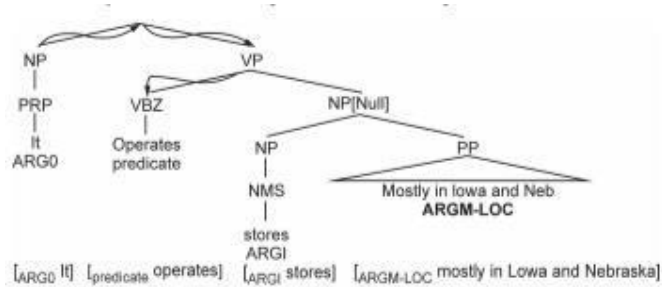
Fig. Q.5.1 : Syntax tree for a sentence illustrating the PropBank tags

2. Predicate: This includes identification of predicate lemma.

3. Phrase type: It identifies the syntactic category.

Ex. NP. PP, S, etc. of the constituent to be labeled.

4. Position: This feature identifies if the phase is present before or after the predicate.

5. Voice: This feature indicates if the predicate is in active or passive role. For passive voiced predicates tgrep27 expressions are used on syntax tree.

6. Head word: This feature represents syntactic head of phrase and is calculated using head word table.

7. Subcate gorization: o This expands the predicates parent node in parse tree.

o For e.g. in Fig. Q.5.1, for predicate operates the subcategorization in given as,

VP $\rightarrow$ VBZ – NP

8. Verb clustering

o In certain cases it may happen that hand tagged training data is very limited and for any real time test set the predicates may not be present in the training.

o In these cases the predicate cluster can be formed and it can be used as a feature.

o For example, the verbs like eat, devour, savov are the object which describe food.

o The distance function is used for clustering based on the fact that verbs with similar semantics will have similar objects.

6. Features suggested by Surdea nu et.al. for semantic role labelling problem.

The features suggested by Surdeanu et al. for addressing semantic role labelling problem includes:

1. Content word As head word feature like PP and SBAR are not so informative, some different rule set is used for identification of content word. The rules are shown in Fig. Q.6.1.

H1: if phrase type is PP then select the rightmost child Example: phrase = "in Texas," content word = "Texas"

H2: if phrase type is SBAR then select the leftmost sentence (S*) clause Example: phrase = "that occurred yesterday," content word = "occurred"

H3: if phrase type is VP then if there is a VP child then select the leftmost VP child else select the head word Example: phrase = "had placed," content word = "placed"

H4: if phrase type is ADVP then select the rightmost child, not IN or TO Example:

phrase = "more than," content word = "more"

H5: if phrase type is ADJP then select the rightmost adjective, verb, noun or ADJP Example: phrase = "61 years old," content word = "61"

H6: for all other phrase types select the head word   Example: phrase = "red house," content word = "red"   Fig. Q.6.1: List of content word heuristics

2. Part of speech of the head word and content word

Part Of Speech (POS) tags can increase the performance the decision tree system.

3. Named entity of the content word

o The roles like ARGM-TMP an ARGM-LOG represent TIME or PLACE named entities.

o This information can be represented as set of binary values.

4. Boolean named entity flags: Named entities like PERSON, PLACE, TIME, DATE etc. are considered as features.

5. Phrasal verbs collections: The frequency of verbs and immediately following preposition in computed and it is used as a feature.

7. Features added by Fleischman, kwon and Hovy to their designed system for semantic role labeling problem. ‹

Following features are added:

1. Logical function

o This function take as input external argument object argument and other argument.

o It is computed by some heuristics on syntax tree and is used as feature.

2. Order of frame elements   Relative position of a frame element to other frame elements in a sentence is considered as a feature.

3. Syntactic pattern Heuristics on phrase type and logic function of the constituent is used as a feature.

4. Previous role n th previous role assigned by system for current predicate is considered as a feature.

8. Explain Combinatory Categorical Grammar (CCG). ‹

x In argument identification task, path feature is very important. x But it is difficult to train feature.

x According to the research of Gildea and Hockenmater, due to features from CCG semantic role labeling is improved.

x Consider the example shown in Fig. Q.8.1 which shows CCG of the sentence "London denied plans on Monday".
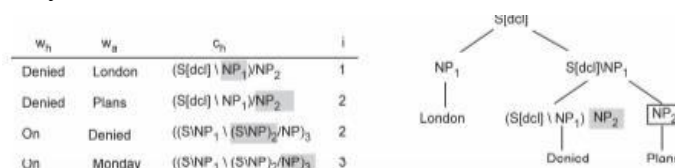

Fig. Q.8.1 : Combinatory category grammar parse

x Three features are proposed:

1. Phrase type: It indicates maximum projection between predicate and dependent word.

2. Categorical path: This feature contain three values

1. Dependent word category.

2. Direction of dependence.

3. Slot in the category filled by dependent word. For example, the path between denied and plans.

3. Tree path: It is a path from dependent word through predicate.

9. Tree-Adjoining Grammar (TAG) is used for semantic role labeling. ‹

x Tree Adjoin Grammar (TAG) is conceptual used by Chen and Rambow and it takes into account the long distance dependencies.

x They used two sets of features to address semantic role labeling problem in predicate argument structure

1. Surface syntactic features.

2. Additional features by extracting TAG from Penn Treebank.

x These additional features include:

1. Supertag path: It is a path features derived from TAG instead of PSG.

2. Supertag: Tree frame feature related to predicate or argument.

3. Surface syntactic role: This feature includes syntactic role of the argument.

4. Surface sub categorization: It is a sub categorization frame feature.

5. Deep syntactic role: This feature includes deep syntactic role of an argument. The values are subject and direct object.

6. Deep sub categorization: It is deep sub categorization frame.

For example, NPO $\rightarrow$ NP1 for transitive verb.

7. Semantic sub categorization: In addition to semantic sub categorization frame, semantic role information is used.

10 Explain problem of semantic role labeling on dependency tree.

Ans.: x It is observed that in case of propBank the system performance depends on how exactly the arguments are annotated according to Penn Tree-bank constituents.

x Correct score of labelled is obtained only if they match propBank annotation exactly.

x To address this problem Hacioglu used dependency tree to convert Penn Tree-bank trees to dependency representation.

x The performance is observed to be increased by SF score point then on phrase structure trees.

x Dependency is established between a word called as head and another called as modifier and a dependency tree is generated.

x Each word can modify at most one other word.

x The headword are classified based on the features shown in Table Q.10.1.

Table Q.10.1 : Features used in the baseline system using Minipar parses

Head word	The word representing the node in the dependency tree.
Head word POS	Part of speech of the head word.
POS path	The path from the predicate to the head word through the dependency tree connecting the part of speech of each node in the tree.
Dependency path	Each word that is connected to the head word has a dependency relationship to the word. These are represented as labels on the arc between the words. This feature comprises the dependencies along the path that connects two words.
Voice	Voice of the predicate.
Position	Whether the node is before or after the predicate.

• In Fig. Q.10.1, we can see the representation of word kick according to phrase structure grammar and minipar parse tree.
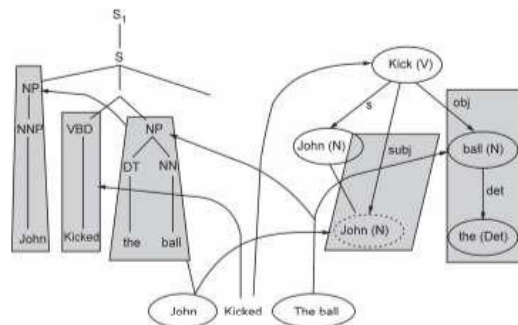
Fig. Q.10.1 : New architecture

11 Chunk based approach for semantic roll labeling. OR Explain base phrase chunks. ‹

x It is observed that in case speech data the chunk representation is faster and robust as compared to phrase reordering.

x In chunk based system each base phrase is classified as B (eginning) of a semantic role, I (inside) a semantic role or o(utside) semantic role.

x This is known as IOB representation.

x In this system the input text is first chunked or divided into base phrases by SVM classifier.

x In the second step, second SVM is trained for assignment of semantic labels to chunks.
x The Table Q.11.1 shows the features used by semantic chunker and Fig. Q.11.1 shows the semantic chunker system.

your roots to success...

Table Q.11.1 : Features used by chunk-based classifier

Words	Words in the chunk
Predicate lemma	The predicate lemma
POS tags	Part of speech of the words in the chunk.
BP positions	The position of a token in a phrase (BP) using the IOB2 representation [e.g. B-NP, I-NP, O]
Clause tags	The tags that mark token positions is a sentence with respect to clauses.
Named entities	The IOB tags of named entities.
Token position	The position of the phrase with respect to the predicate has three values : "before,." "after" and (for the predicate).
Path	Defines a flat path between the token and the predicate.
Clause bracket patterns	
Clause position	A binary feature that identifies whether the token is inside or outside the clause containing the predicate.
Headword suffixes	Suffixes of head words of length 2, 3 and 4.
Distance	Distance of the token from the predicate as a number of base phrases and the distance as the number of VP chunks.
Length	The number of words in a token.
Predicate POS tag	The part of speech category of the predicate.
Predicate frequency	Frequent or rare using a threshold of 3
Predicate BP context	The chain of BPs centened at the predicate within a window of size $-2/+2$.
Predicate POS context	POS tags of words immediately preceding and following the predicate.
Number of predicates	This is the number of predicates in the sentence.

Sales declined 3 % to $ 524.5 million from 539.4 million

Chunk into syntactic base phrases

[NP Sales] (VP declined) [NP 3 %] [PP to] [NP $ 524.5 million] [PP from] [NP $ 539.4 million]

Extract feature

Phrase	Head word	POS	BP	Path	Position	
NP	Sales	NNS	B-NP	NNS → NP → PRED→VBD	b	B-A1
PRED	Declined	VBD	B-VP	-	t	B-V
NP	%	NN	I-NP	NN→NP→PRED→VBD	a	B-A2
PP	To	TO	B-PP	TO→PP→NP→PRED→VBD	a	O
NP	Million	CD	I-NP	CD→NP→PP→NP→PRED→VBD	a	B-A4
PP	From	IN	B-PP	IN→PP→NP→PP→NP→PRED→VBD	a	O
NP	Million	CD	I-NP	CD→NP→PP→NP→PP→NP→PRED→VBD	a	B-A3

Generate parse

[ARG1 Sales] (B-V declined) [ARG2 3 %] to [ARG4 $ 524.5 million] from [ARG3 $ 539.4 million]

Fig. Q.11.1 : Sematnic chunker

12 How machine learning algorithms are used to address semantic role labelling problem.

x Some of the high performance approaches to address the problem of semantic role labeling are explained below:

o Variation of SRL algorithm by Gildea and Jurafsky.

o It consists of two steps:

Step 1: Calculation of maximum likelihood probabilities to know that a constituent is an argument based on features. P (argument |Path, predicate) and P (argument | Head, predicate)

Step 2: Assign each constituent having nonzero probability a normalized probability.

- The distribution is shown in Table Q.12.1.

Table Q.12.1 : Distributions used for semantic argument classification, calculated from the features extracted from a Charniak parse

Distributions
P(argument|Predicate)
P(argument|Phrase type, predicate)
P(argument|Phrase type, position, voice)
P(argument|Phrase type, position, voice, predicate)
P(argument|Phrase type, path, predicate)
P(argument|Phrase type, Path, predicate, subcategorization)
P(argument|Head word)
P(argument|Head word, predicate)
P(argument|Head word, phrase type, predicate)

- Some researchers used decision tree classified algorithm.
- Fleischman and Hovy used FrameNet corpus using maximum entropy framework.
- Pradhan et.al. used SVM.
- Gildea and Palmer used posterior probabilities using different feature sets and does interpolation of estimates.
- The comparison of those classifiers are shown in Table Q.12.2.

Table Q.12.2 : Argument classification using same features but different classifiers

Classifier	Accuracy (%)
SVM (Pradhan et.al.) [120]	88
Decision tree (Surdeanu et.al.) [118]	79
Gildea and Palmer [131]	77

13 What are the different strategies to handle the limitation of trealing semantic role labelling as a series of independent argument classification. ‹

Following are some of the strategies:

R 1. Disallowing overlaps:

x Every constituent is classified independlly.

x It may happen that overlapping constituents are assigned same argument type as shown in the below example.

But [ARG0 nobody] [Predicate knows] [ARG1 at what level] [ARG1 the features and stocks, will open today]

x As overlapping arguments are not allowed in propBank this problem can be solved by choosing the argument for which SVM has highest confidence based on classification probabilities and labeling others as null.

2. Argument sequence information:

x This assumes that to increase performance to argument tagger predicate can instantiate a certain set of arguments.

x Argument ordering information is retained and predicate is considered as a part of argument.

x This is accomplished by first converting raw, SVM scores to probabilities and then crediting argument lattice using a best hypothesis.

x In the next step viterbi search is performed according to the probabilities assigned by sigmoid and maximum likelihood path is obtained.

3. Feature performance: x It is observed that each feature and useful for every task. x Then efficiency depends on classification paradigm. x Table Q.13.1 shows effect of each feature on argument classification and identification.

Table Q.13.1 : Effect of each feature on the argument classification task and argument identification task when added to the baseline system. An asterisx indicates that the improvement is statistically significant

Features	Argument classification	Argument identification		
	A	P	R	F_1
Baseline [120]	87.9	93.7	88.9	91.3
+ Named entities	88.1	93.3	88.9	91.0
+ Head POS	*88.6	94.4	90.1	*92.2
+Verb cluster	88.1	94.1	89.0	91.5
+ Partial path	88.2	93.3	88.9	91.1
+ Verb sense	88.1	93.7	89.5	91.5
+ Noun head PP (only POS)	*88.6	94.4	90.0	*92.2
+ Noun head PP(only head)	*89.8	94.0	89.4	91.7
+ Noun head PP (both)	*89.9	94.7	90.5	*92.6
+ First word in constituent	*89.0	94.4	91.1	*92.7
+ Last word in constituent	*89.4	93.8	89.4	91.6
+ First POS in constituent	88.4	94.4	90.6	*92.5
+ Last POS in constituent	88.3	93.6	89.1	91.3
+ Ordinal constituent pos. concat.	87.7	93.7	89.2	91.4
+ Const. tree distance	88.0	93.7	89.5	91.5
+ Parent constituent	87.9	94.2	90.2	*92.2
+ Parent head	85.8	94.2	90.5	*92.3
+ Parent head POS	*88.5	94.3	90.3	*92.3
+ Right sibling constituent	87.9	94.0	89.9	91.9
+ Right sibling head	87.9	94.4	89.9	*92.1
+ Right sibling head POS	88.1	94.1	89.9	92.0

Features	Argument classification	Argument identification		
	A	P	R	F_1
+ Left sibling constituent	*88.6	93.6	89.6	91.6
+ Left sibling head	86.9	93.9	86.1	89.9
+ Left sibling head POS	*88.8	93.5	89.3	91.4
+ Temporal cue words	*88.6	-	-	-
+ Dynamic class context	88.4	-	-	-

❏ 4. Feature salience :

- Relative contribution of different feature sets as shown in Table Q.13.2 is a contributing factor in system performance.

Table Q.13.2 : Performance of various feature combinations on the task of argument classification

Features	Accuracy
All features [120]	91.0
All except path	90.8
All except phrase type	90.8
All except HW and HW-POS	90.7
All except all phrases	*83.6
All except predicate	*82.4
All except HW and FW and LW info.	*75.1
Only path and predicate	74.4
Only path and phrase type	47.2
Only head word	37.7
Only path	28.0

- The system performance decreases one feature is left at a time.
- In Table Q.13.3, the importance of features on argument identification is shown.

Table Q.13.3 : Performance of various feature combinations on the task of argument identification

Features	P	R	F_1
All features [120]	95.2	92.5	93.8
All except HW	95.1	92.3	93.7
All except predicate	94.5	91.9	93.2
All except HW and FW and LW info.	91.8	88.5	*90.1
All except path and partial path	88.4	88.9	*88.6
Only path and HW	88.5	84.3	86.3
Only path and predicate	89.3	81.2	85.1

5. Feature selection:

x Feature selection strategy is important as it plays different roles in argument classification and argument identification.

x If named entity features are added argument identification task is detoriated whereas argument classification task is improved.

14 Explain how size of training data effects performance of classifier in supervised learning method. ‹

x As per the research done by pradhan et. al the effect of the amount of training data identification and classification of arguments is shown in Fig. Q.14.1.

x The topmost curve in the figure indicates F1 score variation for argument identification task.

x Third curve indicates F1 score on argument identification and classification.

x After 10,000 examples a plateau is reached indicating that only tagging more amount of data will not help to achieve the performance.

15 How to overcome parsing errors in predicate argument structure. ‹

x It has been observed by the researches pradhan et.al. that argument identification is a problem which affects overall systems performance.

x These errors are caused by the fact that

1. Failure of syntactic analyzer in providing constituents mapping to correct arguments.
2. Failure of a system to identify the constituents corresponding to semantic roles.

x These errors are overcomed by following two techniques

1. Combining parses from different syntactic representations.

2. Use of best parses.

16 Explain how semantic role labeling is done for nominal predicates or normalizations.

In a sentence consists of various words consisting of different part of speech tags.

x So it is along with verbs it is also necessary to identify arguments of predicates like nominal, adjectival and pre-positional.

x One way to handle nouns is nominalization i.e. converting a verb into abstract noun.

x Consider the example of nominalization as shown in Fig. Q.16.1.

She complained **about the attack**

She made an official complaint **about the attack**   Nominalized

sentence John walked **around the university**

John took a walk **around the university**   Nominalized

sentence Fig. Q.16.1: Example of nominalization

x The verbs are make and took.

17 Explain the techniques by which nouns can be adapted by features which are originally deviced for verbs. ‹

Some of the features proposed by Pradhan et. al. are limited as follows:

1. Intervening verb features:

x To realize arguments of nominal predicates supporting verbs play important role.

x Three classes being used are:

1. Verbs by being

2. Light verbs like make, take, have.

3. Verbs with POS starting with VB.

x Three features added are:

1. A binary feature which shows presence of verb between predicates and arguments.

2. Actual word as a feature.

3. Path from constituent to verb in a tree.

For example: [Speaker Amit] makes general [predicate assertions] [Topic about marriage]

   2. Predicate NP expansion rule:

x Lower most NP is located and expansion rule is applied to it.

x This features clusters noun phrases with similar internal structure and helps in finding modifiers.

   3. Is predicate plural: This features tells if predicate is singular or plural.

   4. Genitives in constituent:

x Genitive words are the ones with POS PRP, PRP$ or WP$.)

x This features takes value as true if there is genitive word present.

For example [Speaker Burma's] [Phenomenon oil] [Predicate Search] hits virgin forests.

18 Explain various issues in semantic role labelling faced by the languages other than english. ‹

Following are some of the issues, advantages and disadvantages of the systems. Performing semantic role labelling of the languages other than english:

1. Some language specific features like predicate frame feature for chinese language are beneficial for english processing as well.

2. There are certain features which are present only in some languages.

For example, there are no delimiters to separate the words in chinese language. So to do word segmentation more complex system is needed, where are as english language this task become easy.

3. The language like chinese is morphological poor language so predicates and arguments posses close connection as verbs, nouns and adjectives have similarity.

4. Chinese has more verb types than english. So in corpus of same size as english the instances of chinese verbs are less posing the issue of data sparsely.

5. In chinese language syntactic parsing yields less performance than english so shallow parsers give more promising results than full syntactic parsing.

6. In the arabic language POS categories are more than english or chinese, so it is challenging to handle all these by semantic role labeling system and still the rich structure is not explored.

4.2 Meaning Representation Systems

19 What is deep semantic parsing? Explain various various resources used for deep semantic passing. ‹

x The biggest challenge faced by natural language processing system is the ambiguity which exists at every phase.

x It is challenging task to take input in natural language, remove ambiguity, use word knowledge and understand the context and make machines understand the language effortlessly like humans.

x Still the research is going on resolving various ambiguities of the language and till now the systems are developed for working in specific domains, instead of the generalized models.

x This is known as deep semantic parsing.

x Various resource which are developed in this area include:

R 1. ATIS: x Air Travel Information System (ATIS) is a first system which transformed natural language input information for decision making by and application.

x It takes as an input user speech query about flight information in restricted vocabulary.

x It converts it into SQL query and retrieve information from flight database.

x Semantic information is encoded from hierarchical frame representation.

x The training corpus consists of 774 sceneries, 137 subjects, over 7300 utterances.

x 2900 utterances are categorized are represented with reference answers.

x 600 are tree-banked.

x The example user query in shown in Fig. Q.19.1.

```
FRAME Representation
    SHOW :
    FLIGHTS :
    TIME :
    PART-OF-DAY :
    ORIGIN :
    CITY : Boston
    DEST :
    CITY : San Francisco
    DATE :
    DAY-OF-WEEK : Tuesday

    Natural language representation
    Please show me morning flights from Boston to
    San Francisco on Tuesday
```

Fig. Q.19.1 : A sample user query and its frame representation in the ATIS program

❏ 2. Communicator :

- It is an extension of ATIS system.
- ATIS system was designed for user initiated dialog i.e. answers were provided by machine for uses questions.
- Communication system uses mixed initiative dialog. i.e. it is an interactive system in which the communications happens between system and user. Based on real time information provided by system the uses can plan the itinary.
- Many dialogs were collected by the system and stored in linguistic data consortium.
- The data was also collected Carnegic Mellon University.
- About million words and 1600 dialogs are annotated.

Q.20 How natural language is mapped to the meaning representation systems (SQL query, prolog or domain-specific query). ‹

The different ways by which is a natural language input is mapped to domain specific meaning representation system are follows:

R 1. Rule based:

x Rule based semantic parsing system give better results for ATIS and communicator systems.

x In these the speech recognition errors are handled by an interpreter.

x Meaning units in the sentence are parsed to semantic structure, based on the fact that syntactic explanation is complex than semantic information.

x Due to the dynamic variations in the spontaneous speech, pauses, stutters word order is considered to be less important which results in scattering of meaning units and does not cater to the order required by syntactic parser.

x Phoenix is such a system designed by word's based on recursive transition networks and handcrafted grammar.

x The system takes into account hierarchical frame structure and the values of these frames are adjusted with new piece of information.

x The error rate observed is, Spontaneous speech: Input 13.2% with word error rate 4.4%. Transcript input: 4.4%

R 2. Superoised:

x Rule based systems have following shortcomings:

1. Effects for creation of rules.

2. They are time consuming which affects development of systems.

3. Difficult to maintain.

4. Less scalable in case of complex problem.

x To overcome these difficulties statistical models with hand annotated data can be used.

x But if hand annotated data is not present then the system fails.

x To address this problem Schwartz et. al. deviced end to end superoised statistical learning system for ATIS.

x The system has four components:

1. Semantic parse

2. Semantic frame

3. Discourse

4. Backend

x Human-in-the-loop corrective approach is used for training to obtain more data for supervision**.**

5.1 : Cohesion

Q.1 What is cohesion? ‹

x Typically a high level text documents like for ex. Academic articles are divided into different sub sections like Abstract, Introduction, Methodology, Result and Conclusion. x Automatic detection of all these types is a difficult problem.

x To address this problem the algorithms for discourse segmentation are used.

x The unsupervised discourse segmentation algorithms are based on concept of cohesion. x Cohesion is linking of different textual units by using linguistic devices.

x Lexical cohesion is the cohesion which exists in two units which is indicated by relations between words in those units.

For ex.: Consider the sentence Peel, core and slice – the pears and apples.

Add the fruit to skilled.

x In this sentence lexical cohesion between these two sentences is shown by hypernyn relation between fruit and words pears and apples.

5.2 : Reference Resolution

2 What is reference resolution? ‹

x To understand the concept of reference resolution, lets consider following example. "Lakshmi is studying in 10th standard. She is a very good singer and participated in many music programs. The performance of this 16 years old is also excellent in academics."

x In the above passage there is mention of one person named Lakshmi.

x The linguistic expressions like her, she are used to denote an individual is known as reference. x Reference resolution is a task to determine what entities are referred to by which linguistic expressions.

x Referring expression (for ex. she) is a haliral language expression used to perform reference.

x The referred entity is called as reference (for ex. Lakshmi).

x Reference to an entity which is previously introduced in discourse is called anaphora and the referring expression is called as amaphoric.

For ex. Pronoun, she and 16 years old are anaphoric.

x Two referring expressions used to refere same entity are said to corefer.

x So typically the task of reference resolution involve two tasks:

1. Coreference resolution

2. Pronominal anaphora resolution

x Coreference resolution is the task to find out referring expressions referring to same entity.

x Pronominal anaphora resolution is the task to find out antecedent for single pronoun.

For ex.: antecedent of she is Lakshmi. We need to find out the given a pronoun she, its antecedent is Lakshmi.

x Pronominal anaphora resolution can be considered as subtask of coreference resolution.
x There are various algorithms which can be used for this purpose.

Some of them are:

1. Hobbs algorithm

2. Centering algorithm

3. Log linear algorithm

Language Modeling: Introduction, N-Gram Models, Language Model Evaluation, Parameter Estimation, Language Model Adaptation.

Part: I - Language Modelling

5.3 : N - Gram Models

Q.1 What are n gram models? ‹

x To understand concept of n gram models consider the example sentence. x Please turn your homework…..

x We want to predict the next word which can be 'in', 'over' but definitely it will not the word 'the'. This is known as word prediction and can be done using probabilistic models called as n-gram models.

x In n-gram model from sequence of n-token words the next word is predicted from previous n-1 words.

x It is difficult to compute probability of any word sequence w.

x It can be computed by decomposing it based on chain rule of probality as:

$$P(w) = P(w_1 \dots w_t) = (P)(w_1) \prod_{i=1}^{t} P(w/w_{i-1} w_{i-2} \dots w_2 w_1)$$

But as individual product terms also cannot be computed directly n-gram approximation will be useful.

By considering the assumption of history equivalence class that n−1 are useful for predicting a given word, n-gram model can be defined as :

$$P(w) = \prod_{i=1}^{t} P(w_i | w_{i-1}, \dots, w_{i-n+1})$$

x This is based on markov assumption that current word only depends on n – 1 preceding words and in dependent of all the other given words.

x So n-gram model is also called as (n – 1) - th order Markov model.

x Based on length of n the models can be formalized as: for n = 2 → biagram → considering two word sequence of words ex. "please turn" or "turn your" for n=3 o trigram o considering three word sequence of words ex: "please turn your" or "turn your homework" and so on for n = 4,5, etc.

5.4 : Language Model Evaluation

2 Write a note on language model evaluation. ‹

x The evaluate a performance of language model the best way is to include it in an application and check the performance of that application this is called as extrinsic evaluation.

x But as it is expensive at times, another way of evolution i.e. intrinsic evolution is used.

x In intrinsic evolution a metric is used to quickly evaluate the improvements in language model.

x Two criterias used for intrinsic evaluation of language model are.

R 1. Coverage rate

x It measures percentage of n-grams in test set

x Sometimes there are cases where same unknown words appear they are called as Out Of Vocabulary (OOV) words which cannot be handled by this type.

R 2. Perplexity

x It considers the fact that among two probabilistic models the model which fits the test data is the better one.

5.5 : Parameter Estimation

1. Explain maximum likelihood estimation and smoothing. ‹

x Generally n-gram probabilities are estimated by combining maximum likelihood criterion with parameter smoothing.

x The maximum likelihood estimate can be obtained as

$$P(w_i \mid w_{i-1}, w_{i-2}) = \frac{C(w_i, w_{i-1}, w_{i-2})}{C(w_{i-1}, w_{i-2})}$$

Where $C(w_i, w_{i-1}, w_{i-2})$ is count of trigram w_{i-2}, w_{i-1}, w_i in training data.

Smoothing is the process of flattening the peaks in n-gram probability distribution by redistributing probability mass.

Also zero estimates are replaced by some small nonzero values.

x One of the common smoothing technique is called as back off.

x It splits n-grams whose count in training data fall below predetermined threshold T and also whose count exceed the threshold.

- The backed off probability P_{BO} for w_i given w_{i-1}, w_{i-2} is computed as

$$P_{BO}(w_i \mid w_{i-1}, w_{i-2}) = \begin{cases} d_c P(w_i \mid w_{i-1}, w_{i-2}) & \text{it } C > T \\ d(w_{i-1}, w_{i-2}) P_{BO}(w_i \mid w_{i-1}) & \text{otherwise} \end{cases}$$

Where $C \rightarrow$ count of (w_i, w_{i-1}, w_{i-2})

$d_c \rightarrow$ discounting factor applied to higher order distribution.

your roots to success...

Q.6 Explain Bayesian parameter estimation method.

■ Ans. :

- It is a parameter estimation method in which set of parameters of a model are considered as random variable, which is governed by previous statistical distribution posterior distribution.

- $P(\theta|S)$ is given using Bay's rule as

$$P(\theta|S) = \frac{P(S|\theta) P(\theta)}{P(s)}$$

Where $S \rightarrow$ training sample with sequence of words $w_1 \ldots w_t < P(w_1), \ldots, P(w_k) >$ (where k is vocabulary size)

$\theta \rightarrow$ Set of parameters = for unigram model $P(w_1|h_1 \ldots, P(w_t|h_t)) >$ for n-gram

$P(\theta) \rightarrow$ Prior distribution over different possible values of θ

- A point estimate of θ is done by Maximum a Posteriori (MAP) criteria as follows.

$$\theta_{MAP} = \underset{\theta \in \Theta}{\arg\max} \ P(\theta|s) = \underset{\theta \in \Theta}{\arg\max} \ P(S|\theta) P(\theta)$$

Where Θ is space for possible assignments for θ

- The expected value of θ for sample S is given as :

$$\theta_B = E[\theta|S] = \int_\Theta \theta \, P(\theta|S) \, d\theta$$

$$= \frac{\int_\Theta \theta \, P(S|\theta) P(\theta) \, d\theta}{\int_\Theta P(S|\theta) P(\theta) \, d\theta}$$

Q.7 Explain large scale language models.

x With the increase of monolingual data scaling of language is required to handle sets of bilions or trillions of words.

x In this case exact probability computations and in turn parameter estimation is not feasible.

x So to handle the large data, complete data required for training is divided into partitiones.

x Probabilities derived from each partition are stored in separate physical location.

x The language model server handles this data which is distributed over a cluster of independent nodes.

x Clients request statistics from this server during runtime.

x These models facilitate scalabity for handling large amount of data, also new data can be added dynamically.

x The disadvantage is slow speed due to networking overheads.

x Another variation can be use of large scale.

Distributed language at second pass rescoring stage.
x In this first pass hypothesis is done using smaller language models.
x One more approach is storing large scale, language models in working memory of a single machine.
x The concept of bloom filter is used for this purpose.
x The corpus statistics

5.6 : Language Model Adaptation

Q.8 What is language mode adaption ? Explain various methods for doing language model adaption.

Ans. : Sometimes of certain domains or languages the training data required to train a model is insufficient.

- In such cases language model adaptation techniques are used to design and tune a language model so that when it is ported to a new domain it will perform well even if very less training data is available.

- One of the commonly used method is model interpolation or mixture languages models, in which the language model is trained with small training data one domain and a generic model is trained using out of domain data.

- These two models are interpolated based on equation.

$$P(w_i \mid w_{i-1}, w_{i-2}) = \sum_{m=1}^{M} \lambda_m P(w_i \mid h_m)$$

- In the second method stated by Seymour and rosenfield the given documents are clustered based on their topics and a language model in built for each cluster. The final model is generated by interpolation of these models. This method is called as topic dependent language model adaptation.

- Dynamic self adaption of a language model is done through trigger models.

- Trigger model consider the word combination likely to co-occur, 10 in turn one word can trigger other for ex. In financial news text the words stock and market are more likely to co-occur.

- For this purpose Lantent Semantic Analysis (LSA) and Probabilistic Latent Semantic Analyses (PLSA) is used.

- The formula for dynamic adaption of a model is

$$P(w_i \mid h_i, \tilde{h}_i) = \frac{P(w_i \mid h_i)\, P(w_i, \tilde{h}_i)}{Z(h_i, \tilde{h}_i)}$$

Where $\tilde{h}_i \rightarrow$ global document history in LSA space up to word i.

- Another approach is unsupervised adaption which is most suitable for speech recognition applications.

- Instead of written text it uses output of speech recognizer.

- One of probability estimation scheme for language model adaption is Maximum Posteriori (MAP) adaption.

- In this the counts are separately collected from the generic out of domain (OD) and in domain adaptation (1D) are combined as follows :

$$P(w \mid h) = \frac{C_{OD}(\omega, h) \cdot \in C_{ID}(w, h)}{C_{OD}(h) \cdot \in C_{ID}(h)}$$

Where h and w $\rightarrow$ history and predicted words.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

B. Tech III Year I Semester Examinations
NATURAL LANGUAGE
(Common to CSE, IT)

Time: 3 Hours] [Max. Marks: 75   Answer any five questions   All questions carry equal marks

Q.1 a) List and explain the challenges of morphological models.

b) Discuss the importance and goals of natural language processing. [7 + 8]

Q.2 a) Elaborate the models for ambiguity resolution in parsing.

b) With the help of a diagram, explain the representation of syntactic structure. [8 + 7]

Q.3 Explain the word sense systems in detail. [15]

Q.4 a) Difference between predicate and predicator.

b) Write a short note on meaning representation systems. [8 + 7]

Q.5 a) Elaborate the multilingual and cross-lingual language modeling.

b) Illustrate the discourse cohesion and structure. [8 + 7]

Q.6 Explain the performance of approaches in structure of documents. [15]

Q.7 Describe the cohesion and reference resolution. [15]

Q.8 Explain the following:

a) Semantic parsing with execution

b) Language model adaptation.