Unit-IV

Unit-IV

- Predicate Argument Structure
- Meaning Representation Systems

Predicate-Argument Structure

- Resources
- Systems
- Software

Predicate-Argument Structure

- Shallow semantic parsing or semantic role labeling is the process of identifying the various arguments of predicates in a sentence.
- There has been a debate over what constitutes the set of arguments and what the granularity of such argument label should be for various predicates.

Resources

- FrameNet
- PropBank
- Other Resources

Resources

- We have two important corpora that are semantically tagged. One is FrameNet and the other is PropBank.
- These resources have transformed from rule based approaches to more data-oriented approaches.
- These approaches focus on transforming linguistic insights into features.
- FrameNet is based on the theory of frame semantics where a given predicate invokes a semantic frame initiating some or all of the possible semantic roles belonging to that frame.
- PropBank is based on Dowty's prototype theory and uses a more linguistically neutral view. Each predicate has a set of core arguments that are predicate dependent and all predicates share a set of noncore or adjunctive arguments.

FrameNet

- FrameNet contains frame-specific semantic annotation of a number of predicates in English.
- The process of FrameNet annotation consists of identifying specific semantic frames and creating a set of frame-specific roles called **frame elements**.
- A set of predicates that instantiate the semantic frame irrespective of their grammatical category are identified and a variety of sentences are labelled for those predicates.
- The labeling process identifies the following:
  - The frame that an instance of the predicate lemma invokes
  - The semantic arguments for that instance
  - Tagging them with one of the predetermined set of frame elements for that frame.

FrameNet

- The combination of the predicate lemma and the frame that its instance invokes is called a lexical unit (LU).
- Each sense of a polysemous word tends to be associated with a unique frame.
- The verb “break” can mean fail to observe (a law, regulation, or agreement) and can belong to a COMPLIANCE frame along with other word meanings such as violation, obey, flout.
- It can also mean cause to suddenly separate into pieces in a destructive manner and can belong to a CAUSE_TO_FRAGMENT frame along with other meanings such as fracture, fragment, smash.

FrameNet

- Here the frame **Awareness** is instantiated by the verb predicate **believe** and the noun predicate **comprehension**.

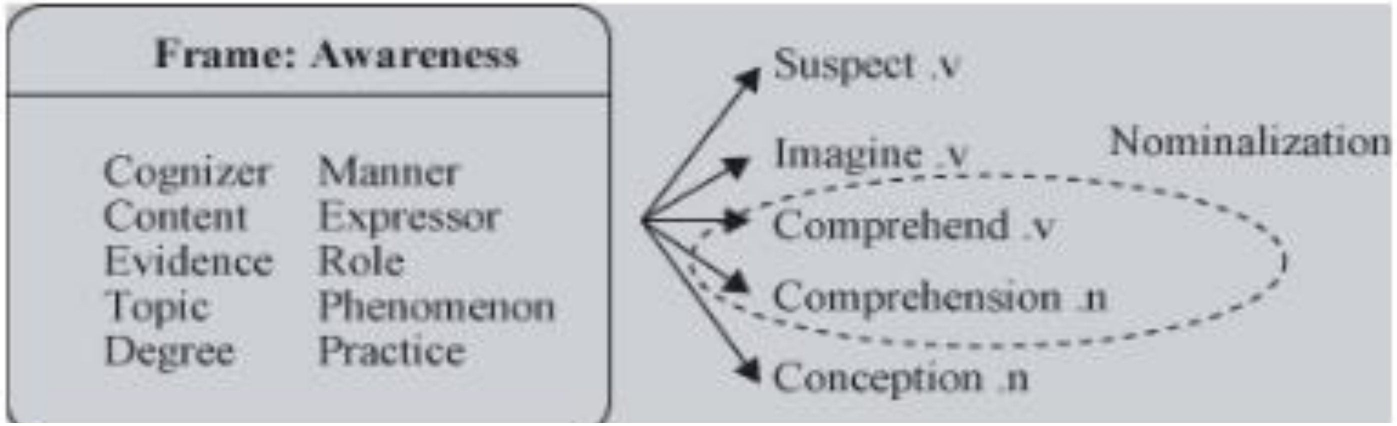


Figure 4–9. FrameNet example

1. [$_{Cognizer}$ We] [$_{Predicate:verb}$ *believe*] [$_{Content}$ it is a fair and generous price]
2. No doubts existed as to [$_{Cognizer}$ our] [$_{Predicate:noun}$ *comprehension*] [$_{Content}$ of it]

FrameNet

- FrameNet contains a wide variety of nominal predicates like:
  - Ultra-nominal
  - Nominals
  - Nominalizations
- It also contains some adjectives and preposition predicates
- The frame elements share the same meaning across the lexical units.
- Example:
  The frame element BODY_PART in frame CURE has the same meaning as the same element in the frame GESTURE or WEARING.

PropBank

- PropBank includes annotations of arguments of verb predicates.
- PropBank restricts the argument boundaries to that of a syntactic constituent as defined in the Penn Treebank.
- The arguments are tagged either:
  - **Core arguments** with labels of type ARGN where N takes values from 0 to 5.
  - **Adjunctive arguments** with labels of the type ARGM-X where X can take values such as TMP for temporal LOC for locative etc.

PropBank

Tag	Description	Examples
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)*

Table 4–2. List of adjunctive arguments in PropBank—ARGMs

PropBank

- Adjunctive arguments share the same meaning across all predicates.
- The meaning of core arguments has to be interpreted in connection with a predicate.

Table 4–1. 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

PropBank

- Let us look at an example from PropBank corpus along with its syntax tree.

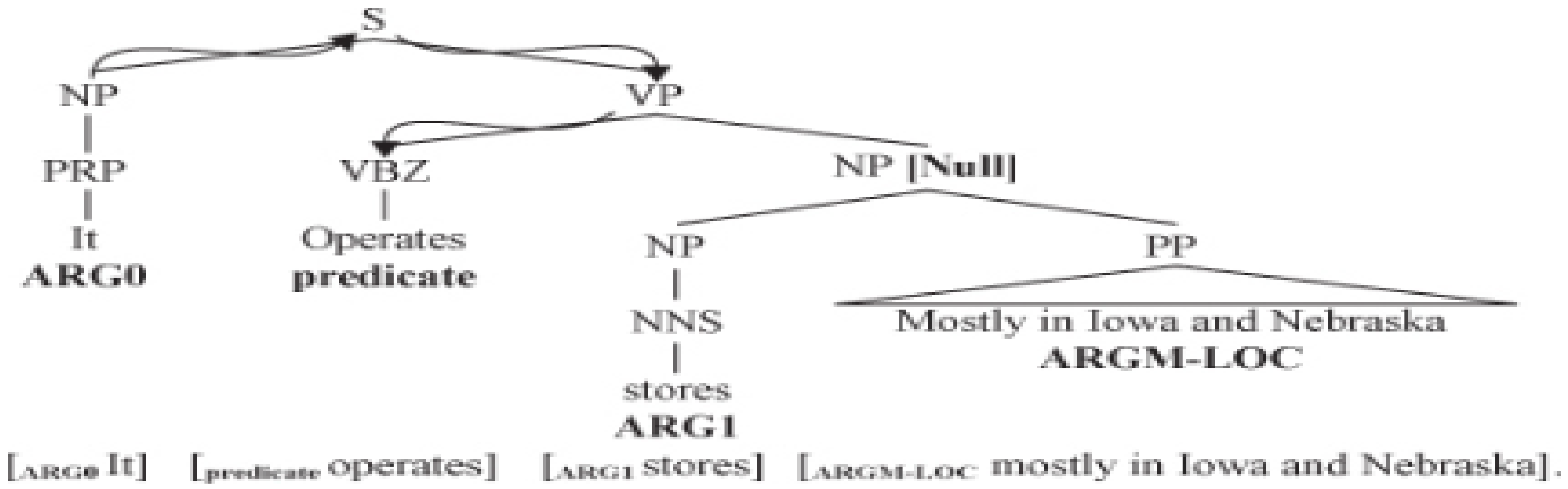


Figure 4–10. Syntax tree for a sentence illustrating the PropBank tags

PropBank

- Most Treebank-style trees have **trace nodes** that refer to another node in the tree but have no words associated with them.
- These can also be marked as arguments.
- Since traces are not reproduced by a usual syntactic parser the community has disregarded them from most standard experiments.
- There are a few disagreements between Treebank and PropBank. In such cases the a sequence of nodes in the tree are annotated as the argument and called as **discontinuous arguments**.

FrameNet Vs Propbank

- An important distinction between FrameNet and Propbank is as follows:
  - In FrameNet we have lexical units which are words paired with their meanings or the frames that they invoke.
  - In Propbank each lemma has a list of different framesets that represent all the senses for which there is a different argument structure.

Other Resources

- Other resources have been developed to aid further research in predicate-argument recognition.
- NomBank was inspired by PropBank.
- In the process of identifying and tagging the arguments of nouns, the NOMLEX (NOMinalization LEXicon) dictionary was expanded to cover about 6,000 entries.
- The frames from PropBank were used to generate the frame files for NomBank.
- Another resource that ties PropBank frames with more predicate-independent thematic roles and also provides a richer representation associated with Levin classes is VerbNet.

Other Resources

- FrameNet frames are also related in the sense that FrameNet’s generation of verb classes is more data driven than theoretical.
- The philosophy of FrameNet and PropBank have propagated to other languages.
- Since the nature of semantics is lingua independent frames can be reused to annotate data in other languages.
- The SALSA project was the first to put this into practice.
- Since FrameNet tags both literal and metaphorical interpretation SALSA project remained close to lexical meaning.
- There are FrameNets in other languages like Japanese, Spanish and Swedish.

Other Resources

- PropBank has inspired creation of similar resources in Chinese, Arabic, Korean, Spanish, Catalan, and Hindi.
- Every new PropBank requires the creation of new set of frame files unlike FrameNet.
- FrameNet and PropBank are not the only styles used in practice.
- Prague Dependency TreeBank tags the predicate argument structure in its tactogrammatic layer on top of dependency structure.
- It also makes a distinction same as core and adjunctive arguments called **inner participants** and **free modifications**.
- The NAIST text corpus is strongly influenced by the traditions in Japanese linguistics.

Systems

- Syntactic Representations
- Classification Paradigms
- Overcoming the Independence Assumptions
- Feature Performance
- Feature Salience
- Feature Selection
- Size of Training Data
- Overcoming Parsing Errors
- Noun Arguments
- Multilingual Issues
- Robustness across Genre

Systems

- Very little research has gone into learning predicate argument structures from unannotated corpora.
- The reason is predicate-argument structure is closer to the actual applications and has been very close to the area of information extraction.
- Early systems in the area of predicate-argument structures were based on heuristics on syntax tree which were rule based.

Systems

- A few of the early systems were:
  - The Absity parser PUNDIT understanding system were among the early rule based systems.
  - One hybrid method for thematic role tagging using WordNet as a resource was introduced.
  - Other notable applications are:
    - Corpus based studies by Manning, Briscoe, and Carroll which seek to derive the subcategorization information from large corpora
    - Pustejovsky which tries to acquire lexical semantic knowledge from corpora

Systems

- A major step in semantic role labelling research happened after the introduction of FrameNet and PropBank.
- One problem with these corpora is significant work goes into creating the frames, that is, in classifying verbs into framesets in preparation for manual annotation.
- Providing coverage for all possible verbs in one or more languages require significant manual effort.
- Green, Dorr and Resnik propose a way to learn the frame structures automatically but the result is not accurate enough to replace the manual frame creation.

Systems

- Swier and Stevenson represent one of the more recent approaches to handling this problem in an unsupervised fashion.
- Let us now look at a few applications after the advent of these corpora.
- Example:

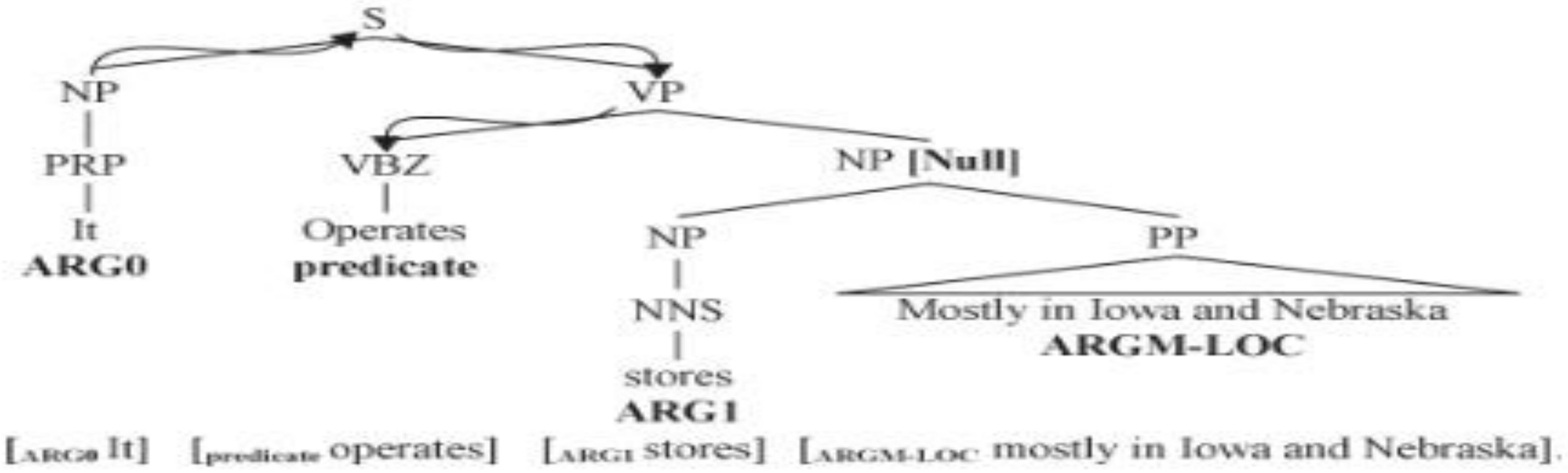


Figure 4–10. Syntax tree for a sentence illustrating the PropBank tags

Systems

- In the example sentence in the previous slide, for the predicate operates, the word “It” fills with the role ARG0, the word “stores” fills the ARG1, and the sequence of words “mostly in Iowa and Nebraska” fills the role ARGM-LOC.
- An ARGN for one predicate need not have similar semantics compared to another predicate.
- FrameNet was the first project that used hand-tagged arguments of predicates in data.
- Gildea and Jurafsky formulated semantic role labeling as a supervised classification problem that assumes the arguments of the predicate.

Systems

- The predicate itself can be mapped to a node in the syntax tree of that sentence.
- They introduced three tasks which can be used to evaluate the system:
  - **Argument Identification**: This is the task of identifying all and only the parse constituents that represent valid semantic arguments of a predicate.
  - **Argument Classification**: Given constituents known to represent arguments of a predicate, assign the appropriate argument labels to them.
  - **Argument identification and classification**: This task is a combination of the previous two tasks where the constituents that represent arguments of a predicate are identified and the appropriate argument label is assigned to them.

Systems

- After parsing each node in the parse tree can be classified as:
  - One that represents a semantic argument (non-null node)
  - One that does not represent any semantic argument (null node)
- The non-null node can further be classified into the set of argument labels.
- In the previous tree the noun phrase that encompasses “mostly in Iowa and Nebraska” is a null node because it does not correspond to a semantic argument.
- The node NP that encompasses “stores” is a non-null node because it does correspond to a semantic argument: ARG1.

Systems

- The pseudo code for a generic semantic role labeling(SRL) algorithm is as follows:

Algorithm 4–3. The semantic role labeling (SRL) algorithm

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 *predicate*
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**

Syntactic Representation

- Phrase Structure Grammar
- Combinatory Categorial Grammar
- Tree Adjoining Grammar

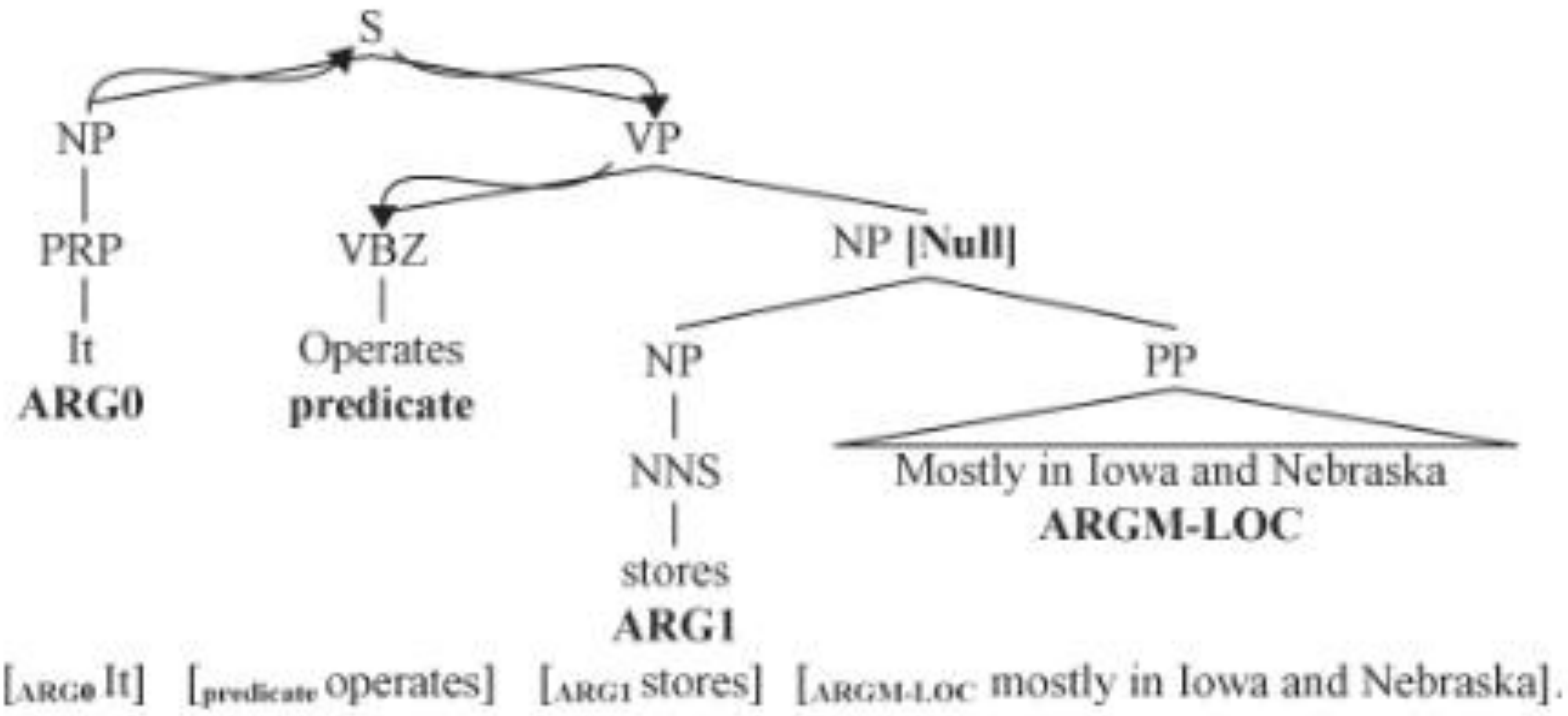
Syntactic Representations

- PropBank was created as a layer of annotation on top of Penn TreeBank style phrase structure trees.
- Gidea and Jurafsky added argument labels to parses obtained from a parser trained on Penn TreeBank .
- Researchers have also used other types of sentence representations to tackle the semantic role labeling problem.
- We now look at a few of these sentence representations and the features that were used to tag text with PropBank arguments.

Phrase Structure Grammar

- FrameNet marks word spans in sentences to represent arguments whereas PropBank tags nodes in a treebank tree with arguments.
- Since the phrase structure representation is amenable to tagging Gildea and Jurafsky introduced the following features:
- **Path**: This feature is the syntactic path through the parse tree from the parse constituent to the predicate being classified.
- For example:
- In the figure in the next slide the path from ARG0 “It” to the predicate “operates” is represented by the string NP$\uparrow S \downarrow VP \downarrow VBZ$.

Phrase Structure Grammar



Phrase Structure Grammar

- **Predicate**: The identity of the predicate lemma is used as a feature.
- **Phrase Type**: This feature is the syntactic category (NP, PP, S, etc.) of the constituent to be labeled.
- **Position**: This feature is a binary feature identifying whether the phrase is before or after the predicate.
- **Voice**: This feature indicates whether the predicate is realized as an active or passive construction. A set of hand written expressions on the syntax tree are used to identify the passive-voiced predicates.

Phrase Structure Grammar

- **Head Word**: This feature is the syntactic head of the phrase. It is calculated using a head word table.
- **Subcategorization**: This feature is the phrase structure rule expanding the predicate's parent node in the parse tree.
- For example:
- In the figure in the previous slide the subcategorization for the predicate "operates" VP$\rightarrow$VBZ-NP.

Phrase Structure Grammar

- **Verb Clustering**:
- This predicate is one of the most salient features in predicting the argument class.
- Gildea and Jurafsky used a distance function for clustering that is based on the intuition that verbs with similar semantics will tend to have similar direct objects.
- For example:
  - Verbs such as eat, devour and savor will occur with direct objects describing food.
- The clustering algorithm uses a database of verb-direct object relations.
- The verbs were clustered into 64 classes using the probabilistic co-occurrence model.

Phrase Structure Grammar

- Surdeanu suggested the following features:
- **Content Word**: Since in some cases head words are not very informative a different set of rules were used to identify a so-called **content** word instead of using the head-word finding rules. The rules that they used are:

```
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"
```

Figure 4–11. List of content word heuristics

Phrase Structure Grammar

- **POS of Head Word and Content Word**: Adding the POS of the head word and the content word of a constituent as a feature to help generalize in the task of argument identification and gives a performance boost to their decision tree-based systems.
- **Named Entity of the Content Word**: Certain roles such as ARGM-TMP, ARGM-LOC tent to contain time or place named entities. This information was added as a set of binary valued features.
- **Boolean Named Entity Flags**: Named entity information was also added as a feature. They created indicator functions for each of the seven named entity types: PERSON, PLACE, TIME, DATE, MONEY, PERCENT, ORGANIZATION.

Phrase Structure Grammar

- **Phrasal Verb Collocations**: This feature comprises frequency statistics related to the verb and the immediately following preposition.
- Fleischman, Kwon, and Hovy added the following features to their system:
- **Logical function**: This is a feature that takes three values external argument, object argument, and other argument and is computed using some heuristic on the syntax tree.
- **Order of Frame Elements**: This feature represents the position of a frame element relative to other frame elements in a sentence.

Phrase Structure Grammar

- **Syntactic Pattern**: This feature is also generated using heuristics on the phrase type and the logical function of the constituent.
- **Previous Role**: This is a set of features indicating the nth previous role that had been observed/assigned by the system for the current predicate.

Phrase Structure Grammar

- Pradhan suggested using the following additional features:
- **Named Entities in Constituents**:
  - Named entities such as location and time are important for the adjunctive arguments ARGM-LOC and ARGM-TMP.
  - Entity tags are also helpful in cases where head words are not common.
  - Each of these features is true if its representative type of named entity is contained in the constituent.
- **Verb Sense Information**:
  - The arguments that a predicate can take depend on the sense of the predicate.
  - Each predicate tagged in the PropBank corpus is assigned a separate set of arguments depending on the sense in which it is used.
  - This is also known as the **frameset ID.**

Phrase Structure Grammar

- The table below illustrates the argument sets for a word. Depending on the sense of the predicate "talk" either ARG-1 or ARG-2 can identify the hearer.

talk.01		talk.02	
Tag	Description	Tag	Description
ARG0	Talker	ARG0	Talker
ARG1	Subject	ARG1	Talked to
ARG2	Hearer	ARG2	Secondary action

- Verb sense information extracted from PropBank is added by treating each sense of a predicate as a distinct predicate which helps performance.
- This disambiguation of PropBank framesets can be performed at a very high accuracy.

Phrase Structure Grammar

- **Noun head of prepositional phrases**:
  - For instance “in the city” and “in few minutes” both share the same head word “in”.
  - The former is ARG-LOC whereas the latter is ARG-TMP.
  - The head word of the prepositional phrase is replaced by the first noun phrase inside the prepositional phrase.
  - The prepositional information is retained by appending it to the phrase type. (Ex PP-IN)
- **First and Last Word/POS in Constituent**: Some arguments tent to contain discriminative first and last words, so these were used along with their POS as four new features.

Phrase Structure Grammar

- **Ordinal Constituent Position**: This feature avoids false positives where constituents far away from the predicate are spuriously identified as arguments.
- **Constituent Tree Distance**: This is finer way of specifying the already present position feature, where the distance of the constituent from the predicate is measured in terms of the number of nodes that need to be traversed through the syntax tree to go from one to the other.
- **Constituent relative features**: These are features representing the constituent type, head word and head word POS of the parent, and left and right siblings of the constituent in focus. This is added for robustness and to improve generalization.

Phrase Structure Grammar

- **Temporal cue words**: Several temporal cue words are not captured by the named entity tagger and were therefore added as binary features indicating their presence. (Temporal words specify order)
- **Dynamic class context**: In the task of argument classification, there are dynamic features that represent the hypothesis of at most the previous two non-null nodes belonging to the same tree as the node being classified.

Phrase Structure Grammar

- **Path generalizations**:
- The path is one of the most salient feature for the argument identification.
- This is also the most data-sparse feature.
- To overcome this problem the path was generalized in several different ways
- **Clause-based path variations**: Position of the clause node (S,SBAR) seems to be an important feature in argument identification. Experiments were done with four clause-based path feature variations.
    - Replacing all the nodes in the path other than clause nodes with an (*).
    - Retaining only the clause nodes in the path.
    - Adding a binary feature that indicates whether the constituent is in the same clause as the predicate.
    - Collapsing the nodes between S nodes.

Phrase Structure Grammar

- **Path n-grams**: This feature decomposes a path into a series of trigrams. For example the path NP↑ $S \uparrow VP \uparrow SBAR \uparrow NP$ ↑VP↓ VBD becomes NP↑ $S \uparrow VP$, $S \uparrow VP \uparrow SBAR$ and so on. Shorter paths can be padded with nulls.
- **Single-Character phrase tags**: Each phrase category is clustered to a category defined by the first character of the phrase label.
- **Path compression**: Compressing sequences of identical labels into one following the intuition that successive embedding of the same phrase in the tree might not add additional information.
- **Directionless path**: Removing the direction in the path, thus making insignificant the point at which it changes direction in the tree.

Phrase Structure Grammar

- **Partial path**: Using only that part of the path from the constituent to the lowest common ancestor of the predicate and the constituent.
- **Predicate context**: This feature captures predicate sense variations. Two words before and two words after were added as features. The POS of the words were also added as features.
- **Punctuations**: For some adjunctive arguments, punctuation plays an important role. This set of features captures whether punctuation appears immediately before and after the constituent.

Phrase Structure Grammar

- **Feature context**:
- Features of constituents that are parent or siblings of the constituent being classified were found useful.
- There is a complex interaction between the types and number of arguments that a constituent can have.
- This feature uses all the other vector values of the constituents that are likely to be non-null as an added context.

Combinatory Categorial Grammar (CCG)

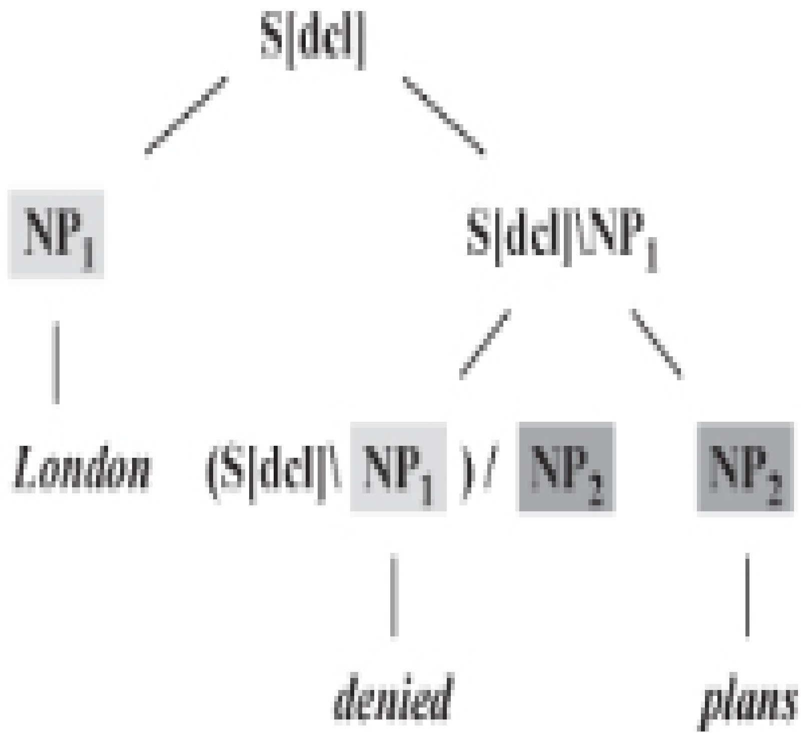
- Though the path feature is important for argument identification task, it is one of the sparse features and difficult to train or generalize.
- Dependency parsers generate shorter paths from the predicate to dependent words in the sentence and can be robust complement to the paths extracted from the PSG parse tree.
- Using features extracted from a CCG representation improves semantic role labeling performance on core arguments.
- CCG trees are binary trees and the constituents have poor alignment with the semantic arguments of a predicate.

Combinatory Categorial Grammar (CCG)

- Let us look at an example of the CCG parse of the sentence "London denied plans on Monday"

London denied plans on Monday

w_h	w_a	c_h	i
denied	*London*	$(S[dcl]\backslash NP_1)/NP_2$	1
denied	*plans*	$(S[dcl]\backslash NP_1/NP_2$	2
on	*denied*	$((S\backslash NP_1\backslash(S\backslash NP)_2/NP)_3$	2
on	*Monday*	$((S\backslash NP_1\backslash(S\backslash NP)_2/NP)_3$	3



Combinatory Categorial Grammar

- Gildea and Hockenmaier introduced three features:
- **Phrase type**: This is the category of the maximal projection between the two words, the predicate and the dependent word.
- **Categorial path**: This is a feature formed by concatenating the following three values:
  - Category to which the dependent word belongs
  - The direction of dependence
  - The slot in the category filled by the dependent word
- The path between denied and plans in the previous figure would be:
  - (S[dcl]\NP)/NP←.

Combinatory Categorial Grammar

- **Tree Path**: This is the categorical analogue of the path feature in the Charniak parse based system, which traces the path from the dependent word to the predicate through the binary CCGtree.

Tree-Adjoining Grammar (TAG)

- TAG has an ability to address long-distance dependencies in text.
- The additional features introduced by Chen and Rambow are:
- **Supertag path**: This feature is the same as the path feature seen earlier except that in this case it is derived from a TAG rather than from a PSG.
- **Supertag**: This feature is the tree frame corresponding to the predicate or the argument.
- **Surface syntactic role**: This feature is the surface syntactic role of the argument.
- **Surface subcategorization**: This feature is the subcategorization frame.

Tree-Adjoining Grammar (TAG)

- **Deep syntactic role**: This feature is the deep syntactic role of an argument, whose values include subject and direct object.
- **Deep subcategorization**: This is the deep syntactic subcategorization frame.
- **Semantic subcategorization**: Gildea and Palmer also used a semantic subcategorization frame where, in addition to the syntactic categories, the feature includes semantic role information.
- A few researchers tried to use a tree kernel that identified and selected subtree patterns from a large number of automatically generated patterns to capture the tree context.
- The performance of this automated process is not as good as handcrafted features.

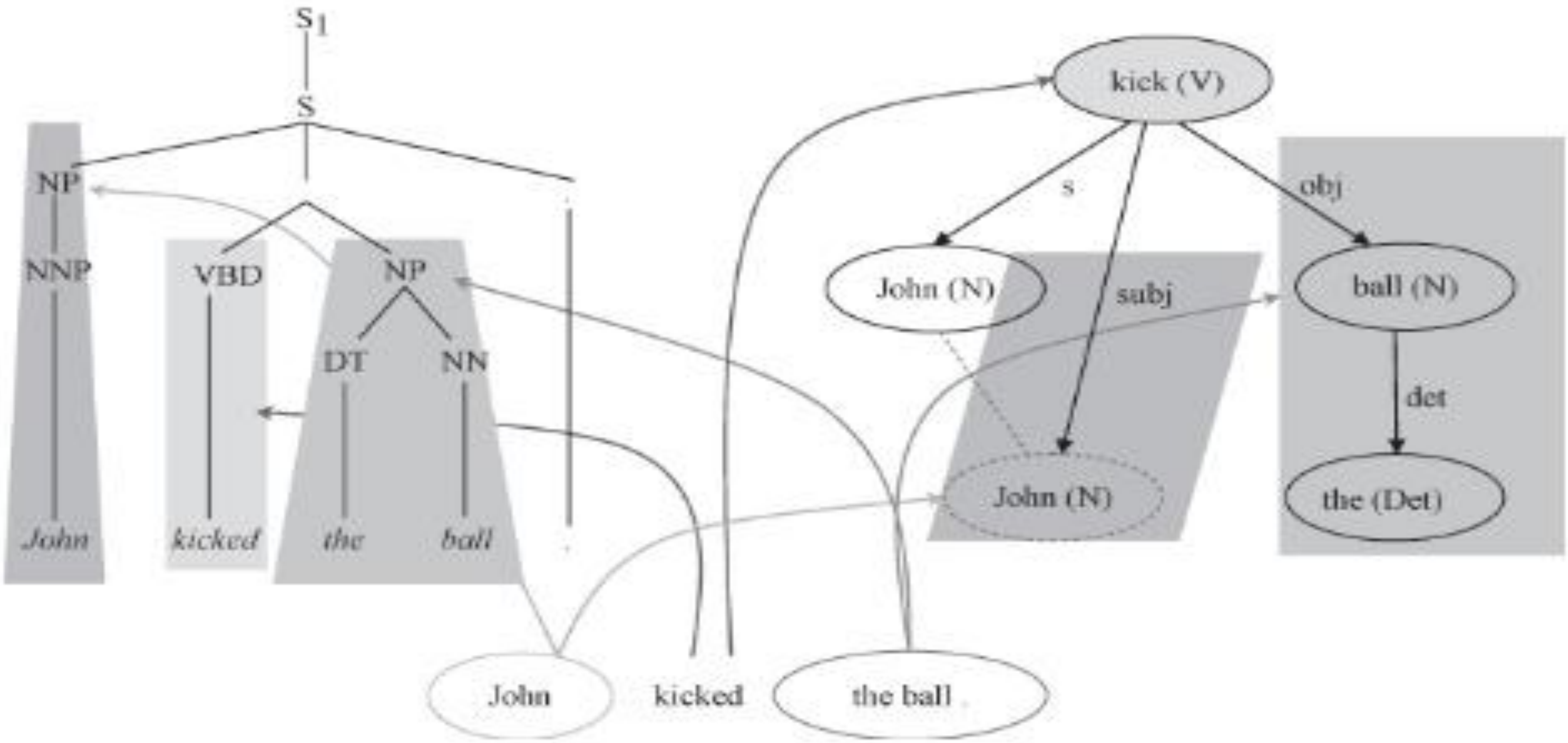
Dependency Trees

- One issue so far is that the performance of a system depends on the exact span of the arguments annotated according to the constituents in the Penn Treebank.
- PropBank and most syntactic parsers are developed in the Penn Treebank corpus.
- They will match the PropBank labeling better than the other representations.
- Algorithms which depend on the relation of the argument head word to the predicate give much higher performance with an F-score of about 85.
- Hacioglu formulated the problem of semantic role labeling on a dependency tree by converting the Penn Treebank trees to a dependency representation.

Dependency Trees

- They used a script by Hwa, Lopez, and Diab and created a dependency structure labeled with PropBank arguments.
- The performance on this system seemed to be about 5 F-score points better than the one trained on the phrase structure trees.
- Parsers trained on Penn Treebank seem to degrade in performance when evaluated on sources other than WSJ.
- Minipar is a rule-based dependency parser that outputs dependencies between a word called head and another called modifier.
- The dependency relationships form a dependency tree.
- The set of words under each node in Minipar's dependency tree form a contiguous segment in the original sentence and correspond to the constituent in a constituent tree.

Dependency Trees



Dependency Trees

- The figure in the previous slide shows how the arguments of the predicate “kick” map to the nodes in a phrase structure grammar tree as well as the nodes in a Minipar parse tree.
- The nodes that represent head words of constituents are the targets of classification.
- They used the features in the following slide.

Dependency Trees

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.

Dependency Trees

- Experiments reported a mismatch of about 8% was introduced in the transformation from Treebank trees to dependency trees.
- A better way to score the performance is to score tags assigned to head words of constituents rather than considering the exact boundaries of the constituents.
- The scores are very good and strengthen the argument for the integration of dependency trees with phrase structure predicate-argument structure.
- Two computational Natural Language Learning (CoNLL) shared tasks were held to further research ways to combine dependency parsing and semantic role labeling.

Dependency Trees - Base Phrase Chunks

- An important question is how much does the full syntactic representation help the task of semantic role labeling?
- How important is it to create a full syntactic tree before classifying the arguments of a predicate?
- A chunk representation can be faster and more robust to phrase reordering as in the case of speech data.
- It was concluded that syntactic parsing helps fill a big gap using chunk based approach by Gildea and Palmer.
- Chunking based systems classify each base phrase as the B(eginning) of a semantic role, I(nside) a semantic role, or O(utside) any semantic role.

Dependency Trees - Base Phrase Chunks

- This is referred to as an IOB representation.
- This system uses SVM classifier to first chunk input text into flat chunks or base phrases, each labeled with a syntactic tag.
- A second SVM is trained to assign semantic labels to the chunks.
- Figure in the next slide shows a schematic of the chunking process.

Dependency Trees - Base Phrase Chunks

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 features

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] ($_{B\text{-}V}$ declined) [$_{ARG2}$ 3 %] to [$_{ARG4}$ $ 524.5 million] from [$_{ARG3}$ $ 539.4 million]

Dependency Trees - Base Phrase Chunks

- The following table lists the features used by the semantic chunker.

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 base phrase (BP) using the IOB2 representation (e.g., B-NP, I-NP, O).
Clause tags	The tags that mark token positions in 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. It 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 centered at the predicate within a window of size $-2/+2$.
Predicate POS context	POS tags of words immediately preceding and following the predicate.
Predicate-argument frames	Left and right core argument patterns around the predicate.
Number of predicates	This is the number of predicates in the sentence.

Classification Paradigms

- Here we focus on the ways in which machine learning has been brought to bear on the problem of semantic role labeling.
- The simplest approaches are those that view semantic role labeling as a pure classification problem.
- Here each argument of a predicate may be classified independent of others.
- A few researchers have adopted the same basic paradigm but added a simple postprocessor that removes implausible analysis, such as when two arguments overlap.

Classification Paradigms

- A few more complicated approaches augment the post processing step to use argument specific language models or frame element group statistics.
- There are more sophisticated approaches to perform joint decoding of all the arguments, trying to capture the arguments interdependence.
- These sophisticated approaches have yielded only slight gains because the performance of a pure classifier followed by a simple postprocessor is already quite high.
- Here we concentrate on a current high-performance approach that is very effective.

Classification Paradigms

- Let us look at the process of the SRL algorithm designed by Gildea and Jurafsky. It involves two steps:
- In the first step:
  - It calculates the maximum likelihood probabilities that the constituent is an argument based on two features:
    - P(argument/Path, Predicate)
    - P(argument/Head, Predicate)
  - It interpolates them to generate the probability that the constituent under consideration represents an argument.
- In the second step:
  - It assigns each constituent that has a nonzero probability of being an argument a normalized probability calculated by interpolating distributions conditioned on various sets of features.
  - It then selects the most probable argument sequence

Classification Paradigms

- Some of the distributions they used are as follows:

Table 4–6. 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)

Classification Paradigms

- Surdeanu used a decision tree classifier C5 on the same features as Gildea and Jurafsky.
- Chen and Rambow used decision tree classifier C4.5 algorithm.
- Fleischman and Hovy report results on the FrameNet corpus using a maximum entropy framework.
- Pradhan used SVM for the same and got even better performance on the PropBank corpus.
- The difference in the result between SVM and entropy classifier is very small.

Classification Paradigms

- The following table compares a few argument classification algorithms using same features:

Table 4–7. 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

Classification Paradigms

- SVMs performs well on text classification tasks where data are represented in a high dimensional space using sparse feature vectors.
- Pradhan formulated the semantic role labeling problem as a multiclass classification problem using SVMs.
- SVMs are inherently binary classifiers but multiclass problems can be reduced to a number of binary-class problems using either the pairwise approach or the one versus all (OVA) approach.
- For an N class problem in the pairwise approach, a binary classifier is trained for each pair of the possible N(N-1)/2 class pairs.
- In the OVA approach, N binary classifiers are trained to discriminate each class from a metaclass created by combining the rest of the classes.

Classification Paradigms

- The SVM system can be viewed as comprising two stages:
  - The training stage
  - The testing stage
- The training stage is divided into two stages:
- In the first stage:
  - Filter out the nodes that have a very high probability of being null
  - A binary classifier is trained on the entire dataset
  - Fit a sigmoid function to the raw scores to convert the scores to probabilities.
  - The respective scores for all the examples are converted to probabilities using the sigmoid function
  - Nodes that are most likely null (probability >.90) are pruned from training set.

Classification Paradigms

- In the second stage the remaining training data are used to train OVA classifiers for all the classed along with a null class.
- With this strategy only one classifier has to be trained on all of the data.
- The remaining classifiers are trained on the nodes passed by the filter.
- In the testing stage all the nodes are classified directly as null or one of the arguments using the classifier trained in step 2.
- A variation in this strategy would be to filter all the examples that are null in the first pruning stage instead of just pruning out the high-probability ones.

Classification Paradigms

- On gold-standard Treebank parsers the performance of such a system on the combined task of argument identification and classification is in the low 90s.
- On automatically generated parsers the performance tends to be in the high 70s.

Overcoming the Independence Assumption

- Various post-processing stages have been proposed to overcome the limitations of treating semantic role labeling as a series of independent argument classification steps. Some of these strategies are:
- Disallowing Overlaps
- Argument Sequence Information

Disallowing Overlaps

- Since each constituent is classified independently it is possible that two constituents that overlap get assigned the same argument type.
- Since we are dealing with parse tree nodes overlapping in words always have an ancestor-descendent relationship.
- The overlaps are restricted to subsumptions only.
- Example:

Example 4–1: But [$_{ARG0}$ nobody] [$_{predicate}$ knows] [$_{ARG1}$ at what level [$_{ARG1}$ the futures and stocks will open today]]

Disallowing Overlaps

- Since overlapping arguments are not allowed in PropBank, one way to deal with this issues is to retain one of them.
- We retain the one for which the SVM has the highest confidence based on the classification probabilities.
- The others are labeled Null.
- The probabilities obtained by applying the sigmoid function to the raw SVM score are used as the measure of confidence.

Argument Sequence Information

- One more way of overcoming the independence assumption is the use of the fact that a predicate is likely to instantiate a certain set of argument types to improve the performance of the statistical argument tagger.
- A better approach involves imposing additional constraints in which argument ordering information is retained and the predicate is considered as an argument and is part of the sequence.
- To achieve this we train a trigram language model on the argument sequence .
- We first convert the raw SVM scores to probabilities.

Argument Sequence Information

- After that for each sentence an argument lattice is generated using the n-best hypotheses for each node in the syntax tree.
- A Viterbi search is then performed through the lattice using the probabilities assigned by the sigmoid function.
- The probabilities are assigned as the observed probabilities along with the language model probabilities to find the maximum likelihood path through the lattice such that each node is assigned a value belonging to the PropBank arguments or null.
- The search is constrained in such a way that no two non-null nodes overlap.

Argument Sequence Information

- To simplify the search we must allow only null assignments to nodes having a null likelihood above a threshold.
- It was found that there was an improvement in the core argument accuracy whereas the accuracy of the adjunctive arguments slightly deteriorated.

Feature Performance

- All features are not equally useful in each task.
- Some features add more noise than information in one context than in another.
- Features can vary in efficacy depending on the classification paradigm in which they are used.
- The table in the following slide shows the effect each feature has on the argument classification and argument identification tasks when added individually to the baseline.
- Addition of named entities to the null/non-null classifier degrade the performance in this particular configuration of classifier and features.

Feature Performance

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 const. 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**
+ 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**	–	–	–

Feature Performance

- The reason for this is the combination of two things:
- A significant number of constituents contain name entities but are not arguments of a predicate resulting in a noisy feature for null/non-null classification.
- SVMs don’t seem to handle irrelevant features very well.

Feature Salience

- In analyzing the performance of the system, it is useful to estimate the relative contribution of the various feature sets used.
- The table in the next slide shows argument classification accuracies for combinations of features on the training and test set for all PropBank arguments using Treebank parsers.
- The features are arranged in the order of increasing salience.
- Removing all head word-related information has the most detrimental effect on performance.

Feature Salience

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

Feature Salience

- The table in the following slide shows the feature salience on the task of argument identification.
- In argument classification task removing the path has the least effect on performance.
- In argument identification task removing the path causes the convergence in SVM training to be very slow and has the most detrimental effect on performance.

Feature Salience

Table 4–10. 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

Feature Selection

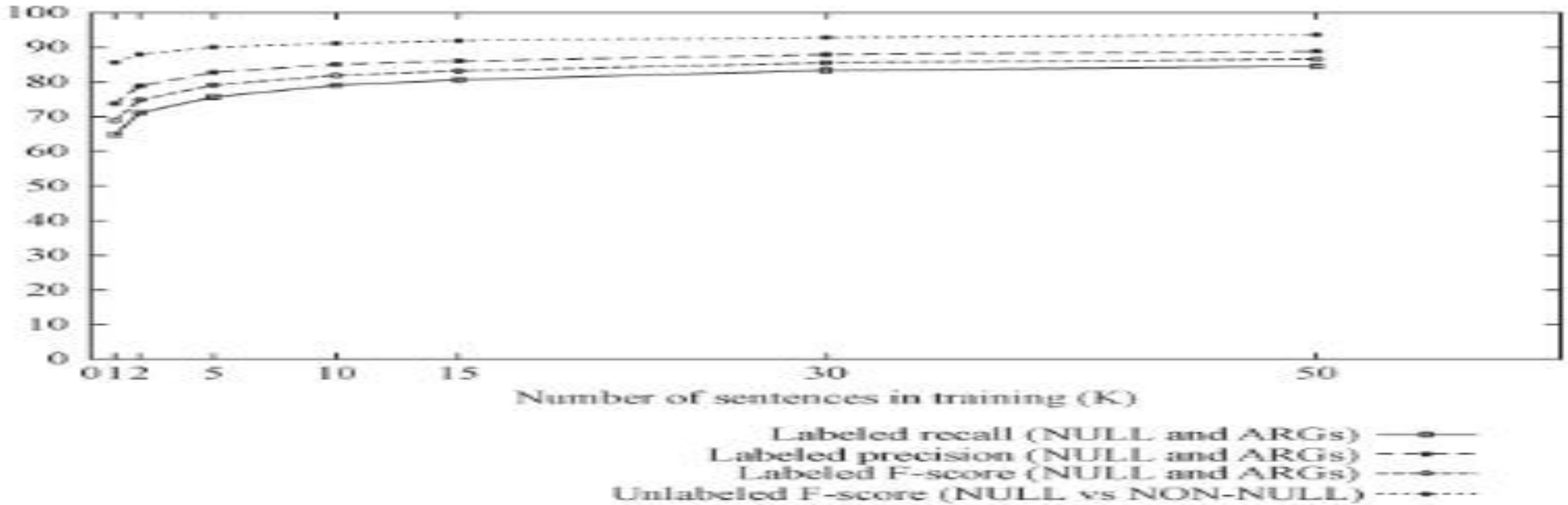
- The fact that adding the named entity feature to the null/non-null classifier has a effect on the performance of the argument identification task.
- The same feature set showed significant improvement to the argument classification task.
- This indicates that a feature selection strategy would be very useful.
- One strategy is to leave one feature at a time and check the performance.
- Depending on the performance the feature is kept or pruned out.

Feature Selection

- One more solution for Feature Selection is to convert the scores that are out put by SVMs to convert into probabilities by fitting a sigmoid.
- The probabilities resulting from either conversion may not properly calibrate.
- In such case the probabilities can be binned and a warping function can be trained to calibrate tehm.

Size of Training Data

- An important concern in any supervised learning method is the amount of training examples required for decent performance of a classifier.
- The results of classifiers trained on varying amount of training data is shown in the following figure.



Size of Training Data

- The first curve from the top indicates the change in F1 score on the task of argument identification alone.
- The third curve indicates the F1 score on the combined task of argument identification and classification.
- We can see that after 10000 examples the performance starts to plateau which indicates that simply tagging more data may not be a good strategy.
- A better strategy is to tag only appropriate new data.
- The fact that both the first and the third curves run parallel to each other tells us that constant loss occurs due to classification errors through the data range.

Overcoming Parsing Errors

- After a detailed error analysis it was found that the identification problem poses a significant bottleneck to improving overall system performance.
- The baseline system’s accuracy on the task of labeling nodes known to represent semantic arguments is 90%.
- Classification performance using Charnaik parses is about 3 F-score points worse than when using treebank parses.
- The severe degradation in argument identification performance for automatic parses was the motivation for examining two techniques for improving argument identification:
  - Combining parses from different syntactic representations – Multiple Views
  - Using n-best parses or a parse forest in the same representation- Broader Search

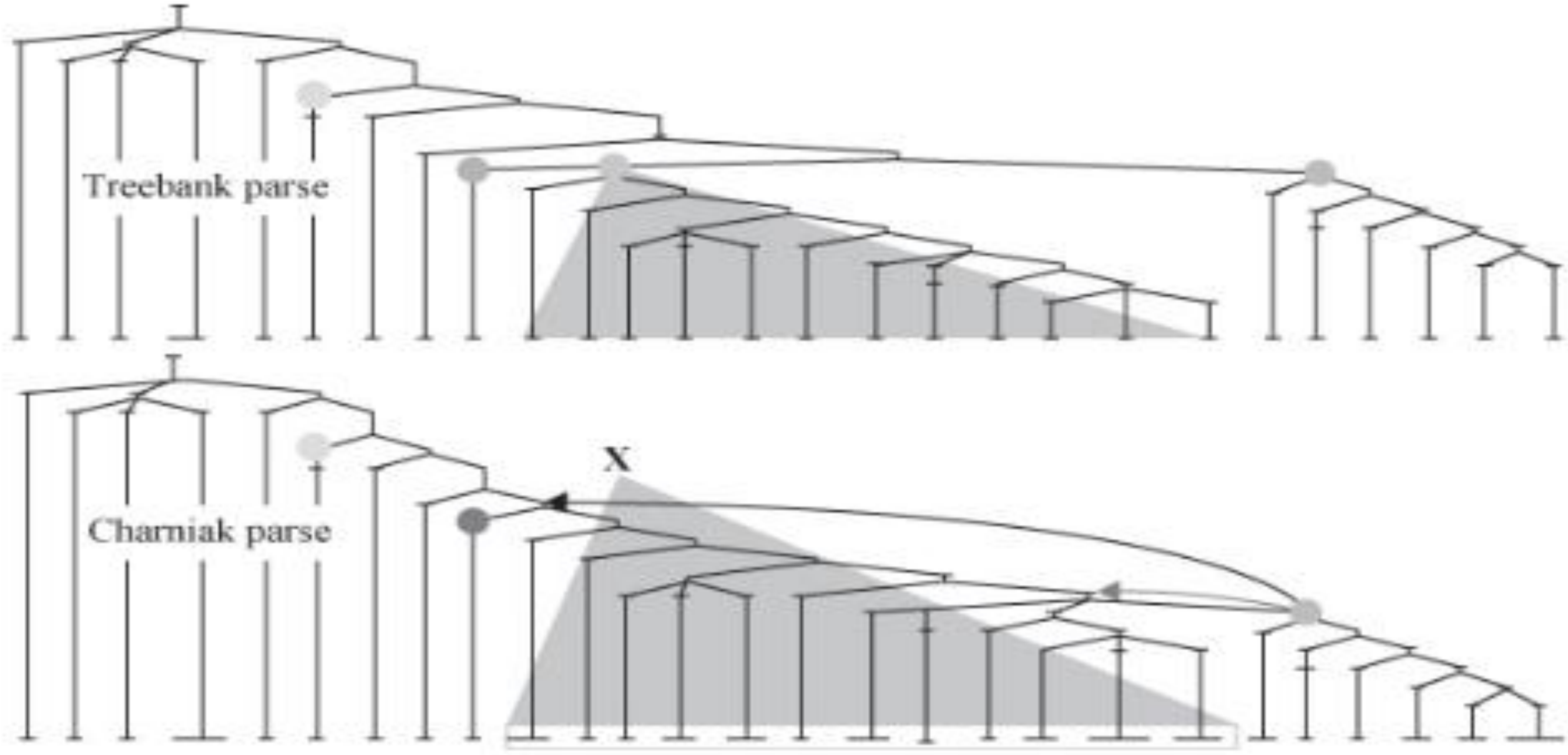
Multiple Views

- Pradhan and others investigated ways to combine hypotheses generated from semantic role taggers trained using different syntactic views:
  - One trained using Charniak parser
  - One on a rule-based dependency parser – Minipar
  - One based on a flat shallow syntactic chunk representation
- They showed that these three views complement each other to improve performance.
- Although the chunk-based systems are very efficient and robust, the systems that use features based on full syntactic parses are generally more accurate.

Multiple Views

- The syntactic parser did not produce any constituent that corresponded to the correct segmentation for the semantic argument.
- Pradhan and others report on a first attempt to overcome this problem by combining semantic role labels produced from different syntactic parses.
- They used features from the Charniak parser, the Minipar parser and a chunk-based parser.
- The main contribution of combining both the Minipar based and the Charniak-based semantic role labeler was improvement on ARG1 in addition to improvement on other arguments as shown in fig in next slide.

Argument Deletions Owing to Parse Error



Multiple Views

- The general framework is to train separate semantic role labeling systems for each of the parse tree views.
- It then uses the role arguments output by these systems as additional features in a semantic role classifier using a flat syntactic view.
- An n-fold cross-validation paradigm is used to train the constituent-based role classifier and the chunk based classifier as in the figure in next slide.

Multiple Views

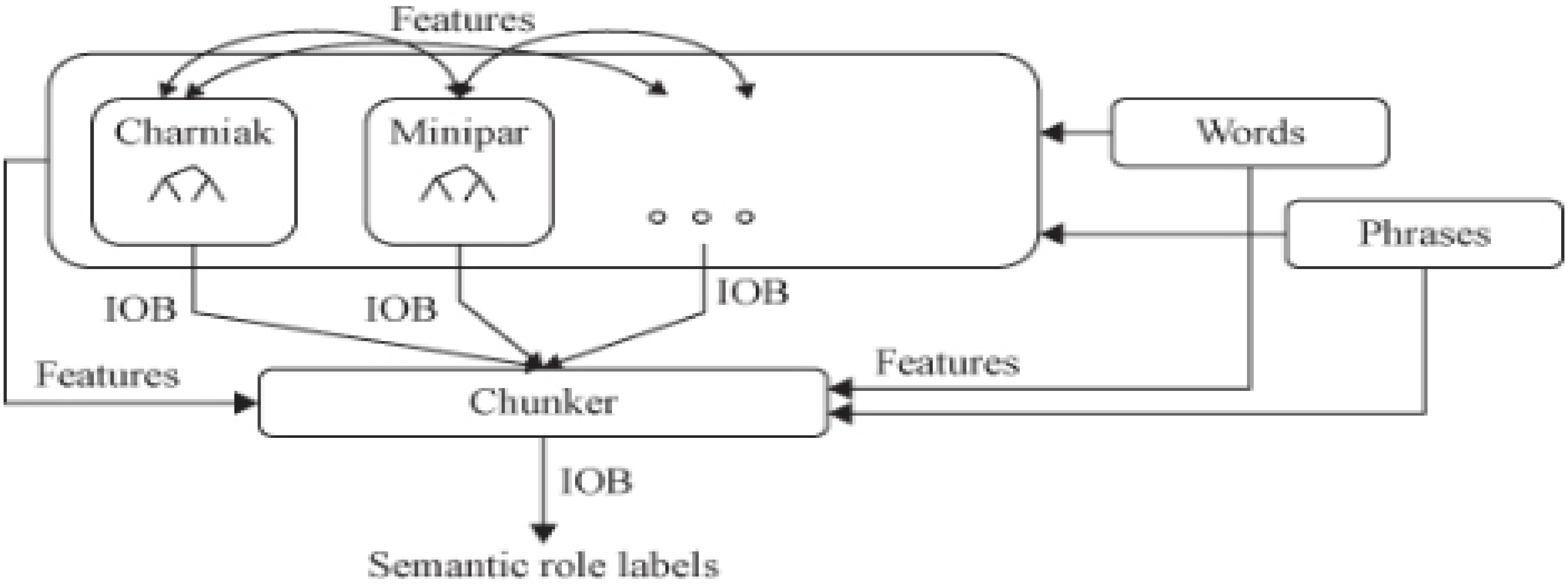


Figure 4–17. New architecture

Broader Search

- One more approach is to broaden the search by selecting constituents in n-best parses or using a packed forest representation which more efficiently represents variations over much larger n.
- Using a parse forest shows an absolute improvement of 1.2 points over single best parses and 0.5 points over n-best parses.

Noun Arguments

- Intervening Verb Features
- Predicate NP expansion rule
- Is predicate plural
- Genitives in constituent
- Verb dominating predicate

Noun Arguments

- **Intervening Verb Features**: Support verbs play an important role in realizing the arguments of nominal predicates. Three classes of intervening verbs are used:
  - Verbs of being
  - Light verbs (a small set of verbs such as make, take, have)
  - Other verbs with part of speech starting with the string VB.
- Three features were added for each:
  - A binary feature indicating the presence of the verb between the predicate and the constituent.
  - The actual word as a feature
  - The path through the tree from the constituent to the verb

Noun Arguments

- The following example illustrates the intuition behind these intervening verb features:
- [$_{\text{Speaker}}$ Leapor] makes general [$_{\text{Predicate}}$ assertions] [$_{\text{Topic}}$ about marriage]
- **Predicate NP expansion rule**: This is the noun equivalent of the verb subcategorization feature used by Gildea and Jurafsky. It represents the expansion rule instantiated by the syntactic parser for the lowermost NP in the tree, encompassing the predicate. This feature tends to cluster noun phrases with a similar internal structure and thus helps find argument modifiers.

Noun Arguments

- **Is predicate plural**: This binary feature indicates whether the predicate is singular or plural, as these tend to have different argument selection properties.
- **Genitives in constituent**: This is a binary feature that is true if there is a genitive word in the constituent, as these tend to be subject/object makers for nominal arguments. The following example helps clarify this notion:
- [$_{\text{Speaker}}$ Burma’s] [$_{\text{Phenomenon}}$ oil] [$_{\text{Predicate}}$ search] hits virgin forests.
- **Verb dominating predicate**: The head word of the first VP ancestor of the predicate.

Multilingual Issues

- Since early research on semantic role labeling was performed on English corpora features and learning mechanisms were explored for English.
- Some special cases of language specific features of other languages proved to be important for the improvement of English systems. Ex: Predicate frame feature introduced for Chinese.
- Some features are language specific and have no parallels in English.
- Special word segmentation models have to be trained in the case of Chinese before parsing or semantic role labeling can begin.

Multilingual Issues

- The morphology poor nature of Chinese blurs the difference between verbs, nouns, and adjectives form a closer connection between the predicates and their arguments.
- Although a similar set of features are useful across languages the specific instantiation of some can differ greatly.
- A particular characteristic of Arabic is its morphological richness.
- There are more syntactic POS categories for Arabic than there are for English or Chinese.
- Unlike English Chinese and Arabic require the training of special models to identify dropped subjects before the predicate-argument structure can be fully realized.

Robustness across Genre

- One important problem with all these approaches is that all the parsers are trained on the same Penn Treebank which when evaluated on sources other than WSJ seems to degrade in performance.
- It has been proved that when we train the system on WSJ data and test on the Brown propositions the classification performance and the identification performance are affected to the same degree.
- This shows that more lexical semantic features are needed to bridge the performance gap across genres.
- Zapirain showed that incorporating features based on selection prefferences provide one way of effecting more lexico-semantic generalization.

Software

- Following is a list of software packages available for semantic role labeling
- **ASSERT** (Automatic Statistical Semantic Role Tagger): A semantic role labeler trained on the English PropBank data.
- **C-ASSERT**: An extension of ASSERT for the Chinese language
- **SwiRL**: One more semantic role labeler trained on PropBank data.
- **Shalmaneser** (A Shallow Semantic Parser): A toolchain for shallow semantic parsing based on the FrameNet data.

Meaning Representation

- Resources
- Systems
- Software

Meaning Representation

- Now we look at the activity which takes natural language input and transforms it into an unambiguous representation that a machine can act on.
- This form will be understood by the machines more than human beings.
- It is easier to generate such a form for programming languages as they impose syntactic and semantic restrictions on the programs where as such restrictions cannot be imposed on natural language.
- Techniques developed so far work within specific domains and are not scalable.
- This is called deep semantic parsing as opposed to shallow semantic parsing.

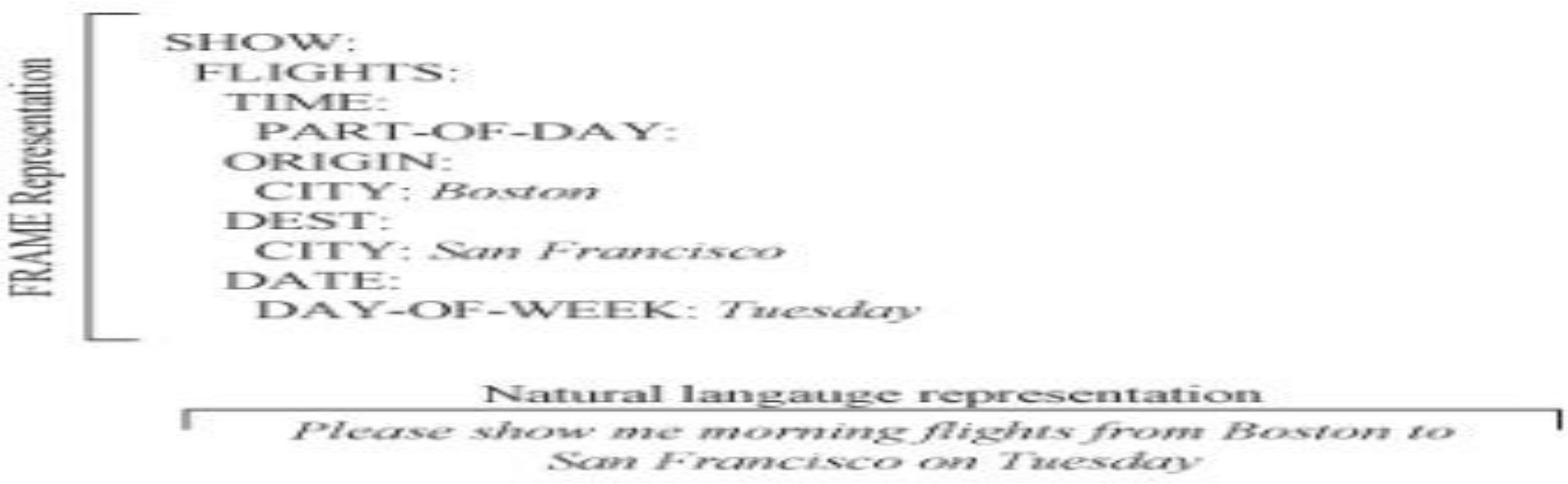
Resources

- A number of projects have created representations and resources that have promoted experimentation in this area. A few of them are as follows:
- ATIS
- Communicator
- GeoQuery
- Robocup:CLang

ATIS

- The Air Travel Information System (ATIS) is one of the first concerted efforts to build systems which build systems to transform natural language into applications to make decisions.
- Here a user query in speech is transformed using a restricted vocabulary about flight information.
- It then formed a representation that was compiled into a SQL query to extract answers from a database.
- A hierarchical frame representation was used to encode the intermediate semantic information.

ATIS



- The training corpus of this system includes 774 scenarios completed by 137 people yielding a total of over 7300 utterances.
- 2900 of then have been categorized and annotated with canonical reference answers.
- 600 of these have also been treebanked.

Communicator

- ATIS was more focus on user-initiated dialog Communicator involved a mixed-initiative dialog.
- Humans and machines were able to have a dialog with each other and the computer was able to present users real time information helping them negotiate a preferred itinerary.
- Many thousands of dialogs were collected and are available through the Linguistic Data Consortium.
- A lot of data was collected and annotated with dialog acts by Carnegie-Mellon university.

GeoQuery

- A geographical database called Geobase has about 800 Prolog facts stored in a relational databse.
- It has geographic information such as population, neighboring states, major rivers, and major cities.
- A few queries and their representations are as follows:
- What is the capital of the state with the largest population?
  - Answer(C,(capital(S,C),largest(P,(state(S),population(S,P)))).
- What are the major cities in Kansas?
  - Answer(C,(major(C),city(C),loc(C,S),equal(S,stated(kansas))))
- The Ge0Query corpus has also been translated into Japanese, Spanish and Turkish.

Robocup:CLang

- RoboCup is an international initiative by the artificial intelligence community that uses robotic soccer as its domain.
- A special formal language Clang is used to encode the advice from the team coach.
- The behaviors are expressed as if-then rules.
- Example:
- If the ball is in our penalty area, all our players except player 4 should stay in our half.
- ((bpos(penalty-area our))(do(player-except our 4)(pos(half our))))

Systems

- Depending on the consuming application the meaning representation can be a SQL query, a Prolog query, or a domain-specific query representation.
- We now look at various ways the problem of mapping the natural language to meaning representation has been tackled.

- Rule Baes
- Supervised

Rule Based

- A few semantic parsing systems that performed very well for both ATIS and Communicator projects were rule-based systems.
- They used an interpreter whose semantic grammar was handcrafted to be robust to speech recognition errors.
- Syntactic explanation of a sentence is much more complex than the underlying semantic information.
- Parsing the meaning units in the sentence into semantics proved to be a better approach.
- In dealing with spontaneous speech the system has to account for ungrammatical instructions, stutters, filled pauses etc.

Rule Based

- Word order becomes less important which leads to meaning units scattered in the sentences and not necessarily in the order that would make sense to a syntactic parser.
- Ward’s system, Phoenix uses a recursive transition networks (RTNs) and a handcrafted grammar to extract a hierarchical frame structure.
- It reevaluates and adjusts the values of the frames with each new piece of information obtained.
- The system had the following error rates:
- 13.2% for spontaneous speech input of which
    - 4.4% speech recognition word-error rate
    - 9.3% error for transcript input

Supervised

- The following are the few problems with rule-based systems:
  - They need some effort upfront to create the rules
  - The time and specificity required to write rules restrict the development to systems that operate in limited domains
  - They are hard to maintain and scale up as the problems become more complex and more domain independent
  - They tend to be brittle
- As an alternative statistical models derived from hand annotated data can be used.
- Unless some hand annotated data is available statistical models cannot deal with unknown phenomena.

Supervised

- During the ATIS evaluations some data was hand-tagged for semantic information.
- Schwartz used that information to create the first end-to-end supervised statistical learning system for ATIS domain.
- They had four components in their system:
  - Semantic parse
  - Semantic frame
  - Discourse
  - Backend
- This system used a supervised learning approach with quick training augmentation through a human in-the-loop corrective approach to generate lower quality but more data for improved supervision.

Supervised

- This research is now known as natural language interface for databases (NLIDB).
- Zelle and Mooney tackled the task of retrieving answers from Prolog database.
- The system tackled the task of retrieving answers from a Prolog database by converting natural language questions into Prolog queries in the domain of GeoQuery.
- The CHILL (Constructive Heuristics Induction for Language Learning) system uses a shift-reduce parser to map the input sentence into parses expressed as a Prolog program.

Supervised

- A representation closer to formal logic than SQL is preferred for CHILL because it can be translated into other equivalent representations.
- It took CHILL 175 training queries to match the performance of Geobase.
- After the advances in machine learning new approaches were identified and existing were refined.
- The SCISSOR (Semantic Composition that Integrates Syntax and Semantics to get Optimal Representation) system uses a statistical syntactic parser to create a Semantically Augmented Parse Tree (SAPT).
- Training for SCISSOR consists of a (natural language, SAPT, meaning representation) triplet.

Supervised

- KRISP (Kernel-based Robust Interpretation for Semantic Parsing) uses string kernels and SVMs to improve the underlying learning techniques.
- WASP (Word Alignment based Semantic Parsing) takes a radical approach to semantic parsing by using state-of-the-art machine translation techniques to learn a semantic parser.
- Wong and Mooney treat the meaning representation language as an alternative form of natural language.
- They used GIZA++ to produce an alignment between the natural language and a variation of the meaning representation language.
- Complete meaning representations are then formed by combining these aligned strings using a synchronous CFG framework.

Supervised

- SCISSOR is more accurate than WASP and KRISP, which benefits from SAPTs.
- These systems also have semantic parsers for Spanish, Turkish, and Japanese with similar accuracies.
- Another approach is from Zettlemoyer and Collins.
- They trained a structured classifier for natural language interfaces.
- The system learned probabilistic categorical grammar (PCCG) along with a log-linear model that represents the distribution over the syntactic and semantic analysis conditioned on the natural language input.

Software

- The software programs available are as follows:
- WASP
- KRISPER
- CHILL