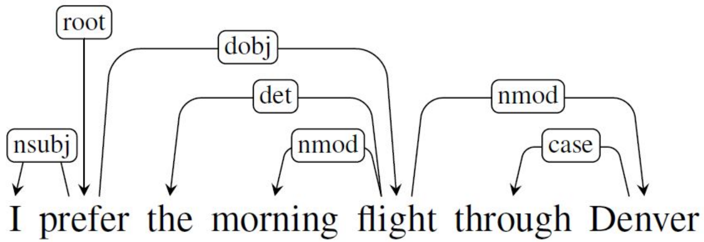
Natural Language Processing (CS5803)

Lecture 9

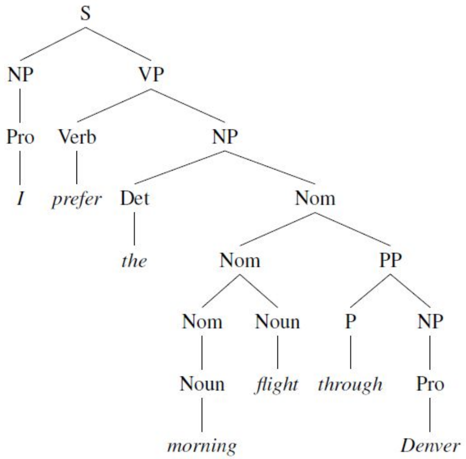
(Parsing - Part 2)

Dependency Grammar

- Syntactic structure of a sentence is described in terms of
  - words (or lemmas) in a sentence
  - set of directed binary grammatical relations that hold among the words
- Relations among the words are expressed with directed, labeled arcs from heads to dependents
- Can contain punctuations. Also, stems and suffixes for some languages
- "root" denotes head of the entire structure
- Also called Typed Dependency Structure
  - As the labels are drawn from a fixed inventory of grammatical relations



Dependency-based analysis



Phrase-structure analysis

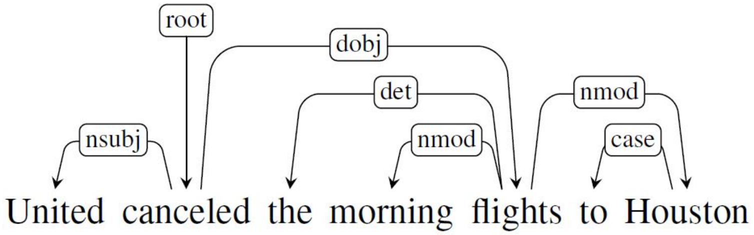
Dependency Parsing Demo: Online resources

- https://corenlp.run/
- https://demo.allennlp.org/dependency-parsing
- https://explosion.ai/demos/displacy

Dependency Relation

- Describes the role that the dependent plays with respect to its head
- **Clausal Relations**
  - Describe syntactic roles with respect to a predicate (often a verb)
- **Modifier Relations**
  - Categorize the ways that words can modify their heads

Clausal Argument Relations	**Description**
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction



- Causal relations NSUBJ and DOBJ identify the subject and direct object of the predicate *cancel*
- NMOD, DET, and CASE relations denote modifiers of the nouns *flights* and *Houston*.

Some Example Dependency Relations

Relation	Examples with *head* and **dependent**
NSUBJ	**United** *canceled* the flight.
DOBJ	United *diverted* the **flight** to Reno.
	We *booked* her the first **flight** to Miami.
IOBJ	We *booked* **her** the flight to Miami.
NMOD	We took the **morning** *flight*.
AMOD	Book the **cheapest** *flight*.
NUMMOD	Before the storm JetBlue canceled **1000** *flights*.
APPOS	*United*, a **unit** of UAL, matched the fares.
DET	**The** *flight* was canceled.
	Which *flight* was delayed?
CONJ	We *flew* to Denver and **drove** to Steamboat.
CC	We flew to Denver **and** *drove* to Steamboat.
CASE	Book the flight **through** *Houston*.

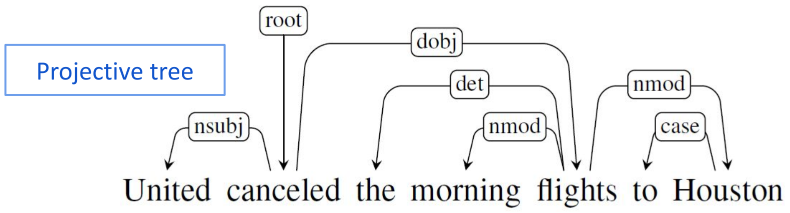
Why Dependency Grammar?

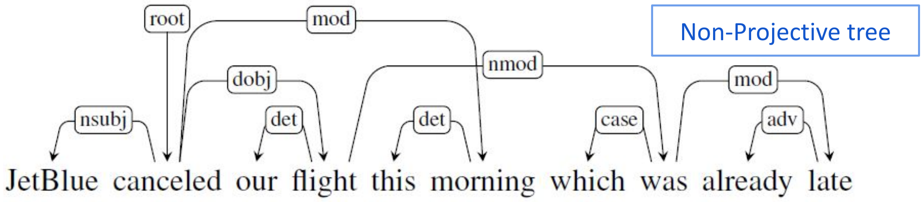
- Abstracts away from word-order information through dependency relations
  - CFG would need a separate rule for each possible place
  - Helpful for morphologically rich languages that have relatively free word order

- Head-dependent relations provide an approximate semantic relationship
  - Useful for applications such as coreference resolution, question answering and information extraction

Dependency Tree

- Directed graph G = (*V*,*A*)
  - *V* : set of words(or punctuation) in a given sentence
  - *A* : set of arcs that captures head-dependent relation
- Properties of Dependency Tree
  - Connected
  - Single designated root node that has no incoming arcs
  - Each vertex has exactly one incoming arc except the root
  - There is a unique path from the root node to each vertex
- Projectivity
  - Dependency tree is projective if it can be drawn with no crossing edges
  - Non-projective trees arise in languages with relatively flexible word order

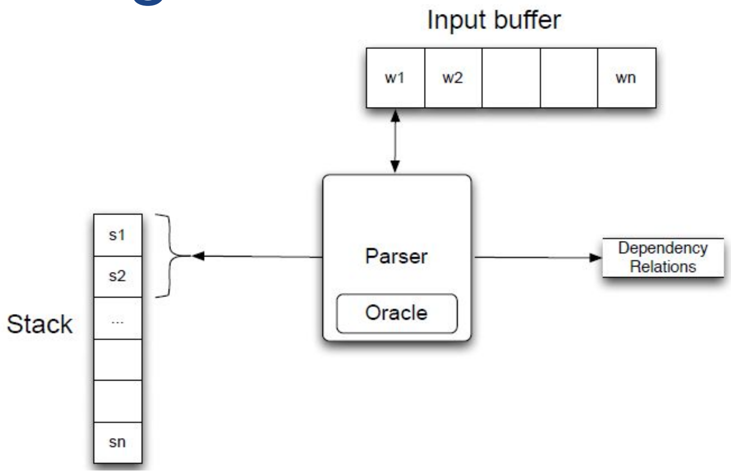




Datasets and related resource: https://universaldependencies.org/

Transition based dependency parsing

- Based on shift-reduce parsing (from Compiler)
- Initial Configuration
  - Stack contains ROOT node,
  - Word list is initialized with the set of the words
- Goal Configuration
  - Empty stack and word list
  - Set of relations represents the final parse
- Actions
  - LEFT ARC
    - head-dependent relation between top and (top-1)
    - remove (top-1) from the stack
  - RIGHT ARC
    - head-dependent relation between (top-1) and top
    - remove top from the stack
  - SHIFT
    - Remove the word from the front of the input buffer and

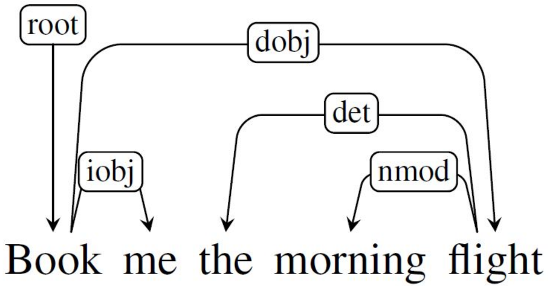



```
function DEPENDENCYPARSE(words) returns dependency tree

  state ← {[root], [words], [] }  ; initial configuration
  while state not final
    t ← ORACLE(state)        ; choose a transition operator to apply
    state ← APPLY(t, state)  ; apply it, creating a new state
  return state
```

Example of transition-based parsing

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	
6	[root, book, the, morning, flight]	[]	LEFTARC	(morning ← flight)
7	[root, book, the, flight]	[]	LEFTARC	(the ← flight)
8	[root, book, flight]	[]	RIGHTARC	(book → flight)
9	[root, book]	[]	RIGHTARC	(root → book)
10	[root]	[]	Done	

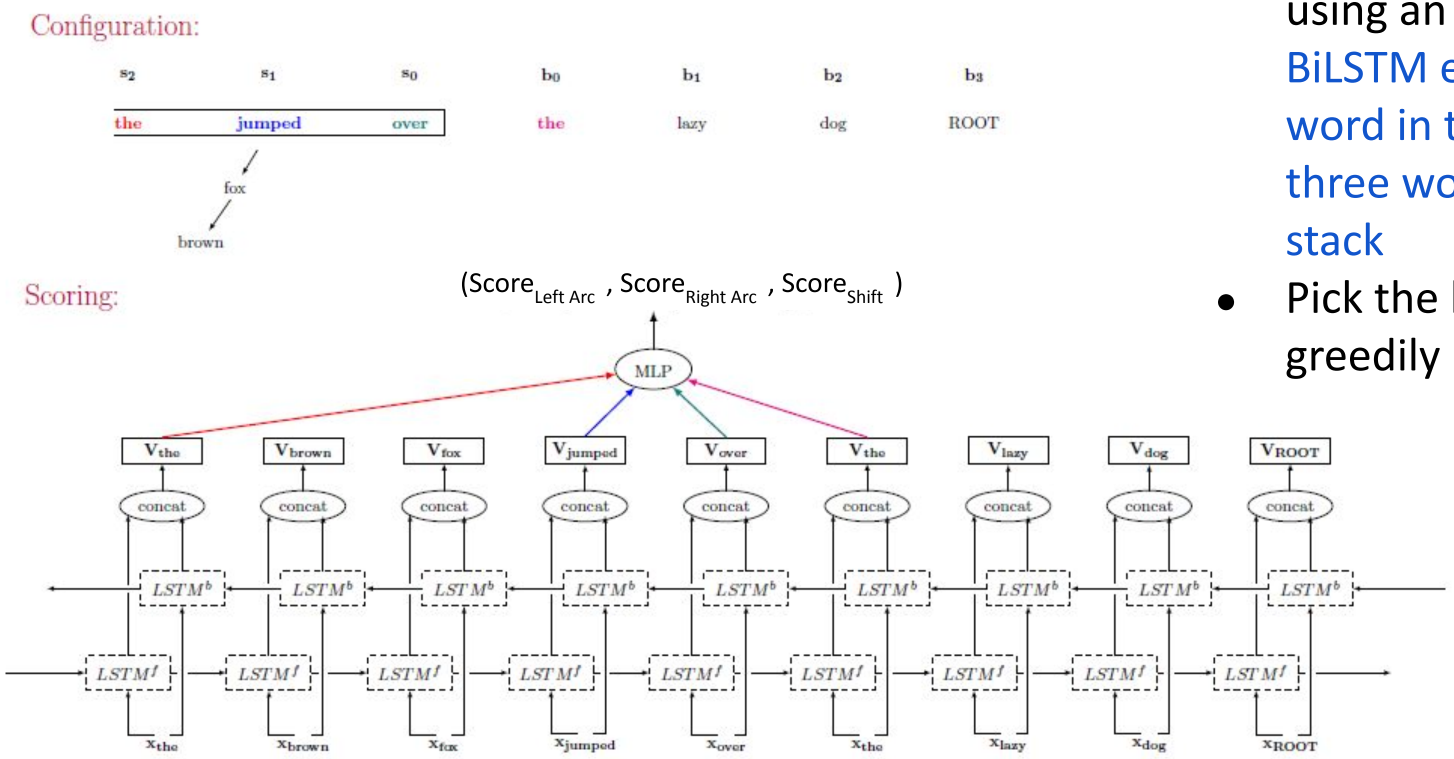


Some comments on transition-based parsing

- Are the sequence of actions unique? NO
  - There may be more than one path that leads to the same result
  - There may be other transitions that lead to different equally valid parses due to ambiguity
- How to generate the labels of the arcs?
  - Parameterize the LEFT_ARC and RIGHT_ARC operators with dependency labels [Eg. LEFT_ARC(NSUBJ) or RIGHT_ARC(DOBJ)]
  - Job of Oracle will be to return the correct action from a large set of actions
- Is Oracle always right? NO
  - Very unlikely to be true in practice
- How to create a good Oracle?
  - Supervised Machine Learning methods

NN approach for transition-based parsing

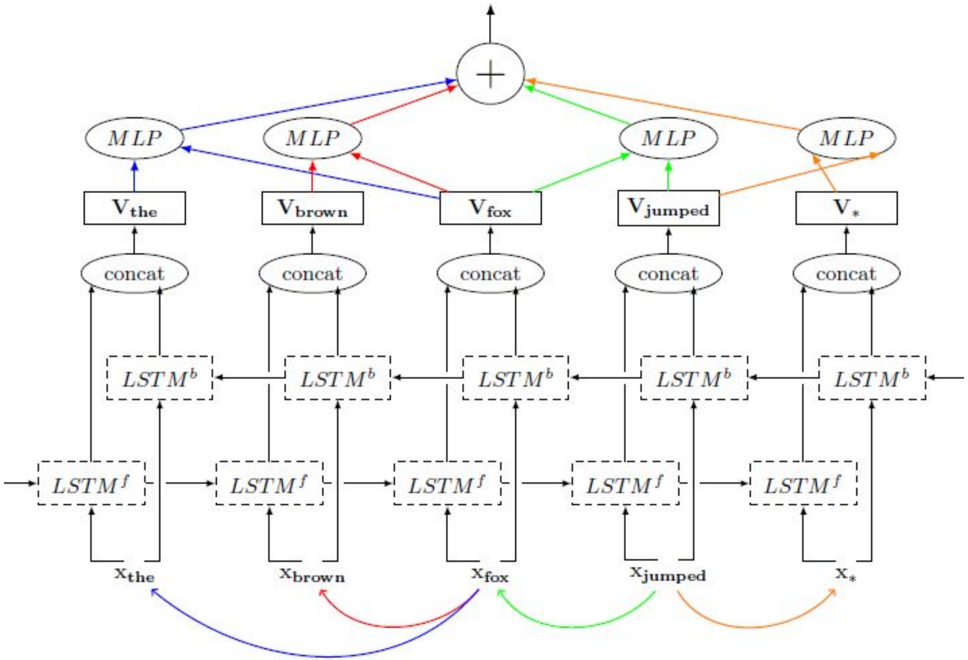
- [Kiperwasser & Goldberg (2016)](#)



- Each transition is scored using an MLP that is fed the BiLSTM encodings of the first word in the buffer and the three words at the top of the stack
- Pick the best transition greedily

NN approach for graph-based parsing

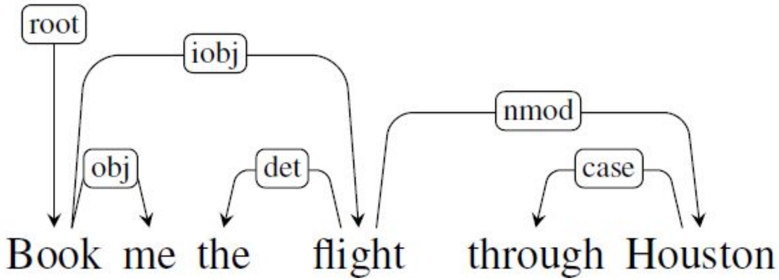
- Kiperwasser & Goldberg (2016)
- Basic Idea
    - Each dependency arc is scored using an MLP that is fed the BiLSTM encoding of the words at the arc's end points
    - Individual arc scores are summed to produce the final score
    - All the MLPs share the same parameters
    - When parsing, compute scores for all possible n^2 arcs and find the best scoring tree
- Label the edges using another classifier (Dozat & Manning, 2016)



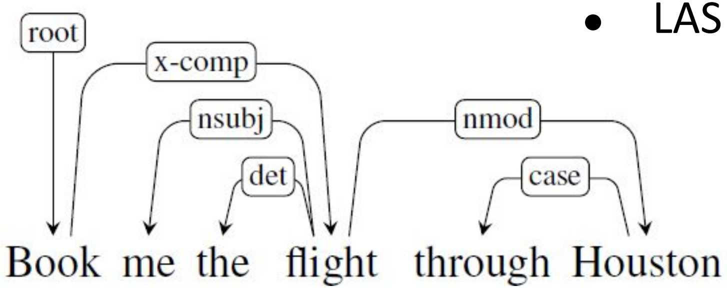
Evaluation of dependency parser

- Unlabeled Attachment Accuracy
  - Percentage of words in input assigned with the correct head
  - Also referred as unlabeled attachment score (UAS)
- Labelled Attachment Accuracy
  - Percentage of words in input assigned with correct head and dependency relation
  - Also referred as labeled attachment score (LAS)

- UAS = 5/6
- LAS = 4/6



Actual Tree

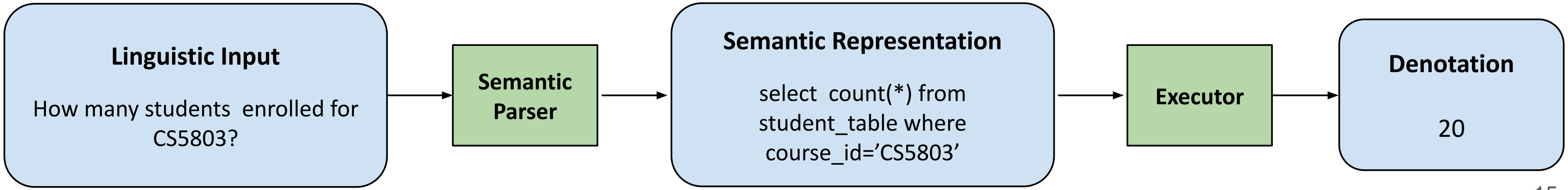


Predicted Tree

Part 3 - Semantic Parsing*

What is Semantic Parsing?

- Semantic parsing is a computation which takes a linguistic input and returns as output a structured, machine-readable representation of its meaning, known as the semantic representation
- Downstream component which consumes the output of the semantic parser is known as the executor
- Output of the executor is known as denotation
- **If we want to understand natural language completely and precisely, we need to do semantic parsing**



Semantic Role Labeling

Semantic Roles

- **Semantics:** deals with meaning

- Consider the following sentences:
    - Last week, Min broke the window with a hammer.
    - The window was broken with a hammer by Min last week
    - With a hammer, Min broke the window last week
    - Last week, the window was broken by Min with a hammer
    - Min broke the window
    - The window broke
    - The window was broken with a hammer

Semantic Roles

Yesterday, Kristina hit Scott with a baseball

Scott was hit by Kristina yesterday with a baseball

Yesterday, Scott was hit with a baseball by Kristina

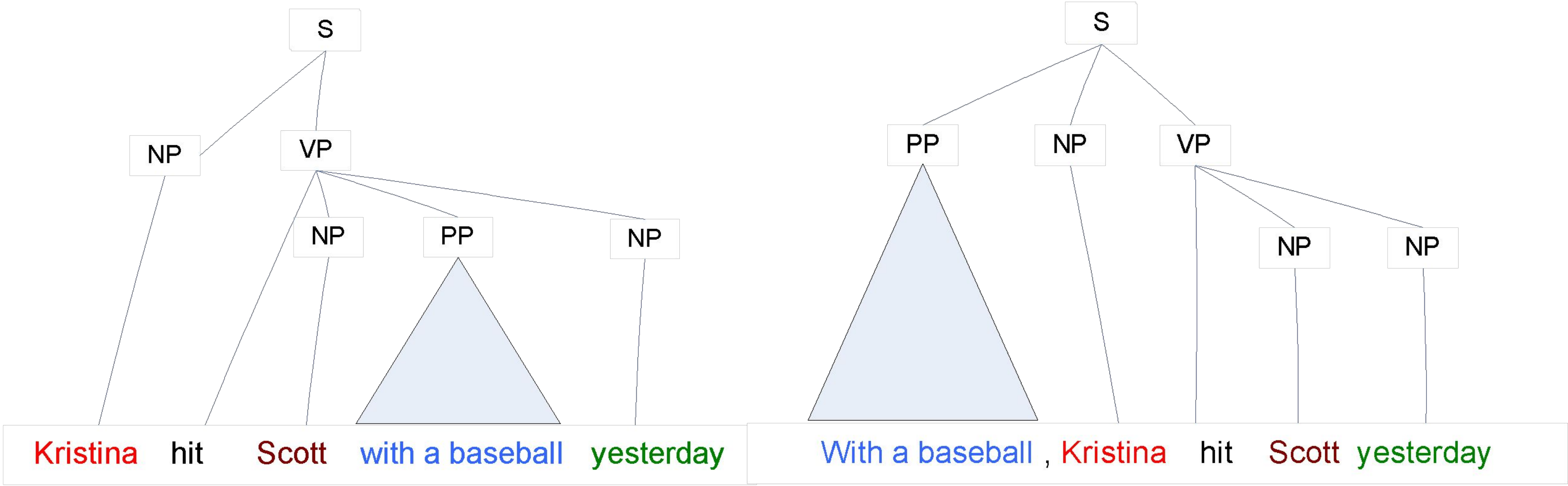
With a baseball, Kristina hit Scott yesterday

Yesterday Scott was hit by Kristina with a baseball

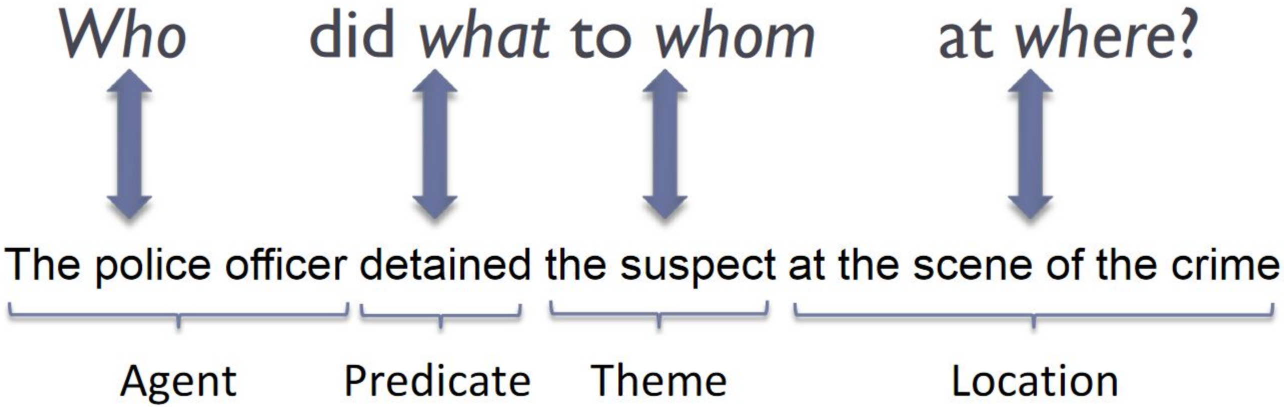
Kristina hit Scott with a baseball yesterday

Agent, hitter	Thing hit	Instrument	Temporal adjunct

Parse trees corresponding to two sentences



Semantic Roles



Semantic Roles

- **Determining**
  - who
  - did what
  - to whom
  - when
  - where
  - why
  - how
- **Uses**
  - Question answering
  - Machine translation
  - Text summarization

Cases (विभक्तयः)	Function	Prepositions	Example
प्रथमा (Nominative)	कर्ता (Subject)	-	देवः अस्मान् रक्षति ।
द्वितीया (Accusative)	कर्म (Object)	To	अहं देवं नमामि ।
तृतीया (Instrumental)	करणम् (Instrument)	By/With/Through	राक्षसाः देवेन ताडिताः ।
चतुर्थी (Dative)	सम्प्रदानम् (Receiver)	To/ For	अहं देवाय दुग्धम् आनयामि ।
पञ्चमी (Ablative)	अपादानम् (Point of separation)	From	अहं देवात् वरान् प्राप्नोमि ।
षष्ठी (Genitive)	सम्बन्धः (Possession/ Relation)	Of/'s	देवस्य कीर्तिः अद्वितीया ।
सप्तमी (Locative)	अधिकरणम् (Location)	In/On/At/Among	सर्वं जगत् देवे एव अस्ति ।
संबोधनम् (Vocative)	सम्बोधनम् (To address someone)	O!	हे देव, रक्ष माम् ।

Case Theory (Fillmore 1968)

- Agent
  - Actor of an action
  - **The musician** performed a new piece
- Patient
  - Entity affected by the action
  - Samantha hurt **her hand**
- Instrument
  - Tool used in performing action
  - Min broke the window **with a hammer**
- Beneficiary
  - Entity for whom action is performed
  - The mother bought ice cream **for the children**
- Source
  - Origin of the affected entity
  - I got the book **from my friend**
- Destination
  - Destination of the affected entity

Thematic Roles

- A typical set:

Thematic Role	Definition	Example
AGENT	The volitional causer of an event	*The waiter* spilled the soup.
EXPERIENCER	The experiencer of an event	*John* has a headache.
FORCE	The non-volitional causer of the event	*The wind* blows debris from the mall into our yards.
THEME	The participant most directly affected by an event	Only after Benjamin Franklin broke *the ice*...
RESULT	The end product of an event	The city built a *regulation-size baseball diamond*...
CONTENT	The proposition or content of a propositional event	Mona asked *"You met Mary Ann at a supermarket?"*
INSTRUMENT	An instrument used in an event	He poached catfish, stunning them *with a shocking device*...
BENEFICIARY	The beneficiary of an event	Whenever Ann Callahan makes hotel reservations *for her boss*...
SOURCE	The origin of the object of a transfer event	I flew in *from Boston*.
GOAL	The destination of an object of a transfer event	I drove *to Portland*.

Thematic Grid

Example usages of "break"

John *broke the window.*
AGENT THEME

John *broke the window with a rock.*
AGENT THEME INSTRUMENT

The rock *broke the window.*
INSTRUMENT THEME

The window broke.
THEME

The window was broken by John.
THEME AGENT

thematic grid, case frame, θ-grid

Break:
AGENT, THEME, INSTRUMENT.

Some realizations:

AGENT/Subject, THEME/Object
AGENT/Subject, THEME/Object, INSTRUMENT/PP_{with}
INSTRUMENT/Subject, THEME/Object
THEME/Subject

Can be used to identify the semantic roles given an input sentence

Verb Alteration

- Doris gave the book to Cary
- Doris gave Cary the book.

- Sequence alternation for some verbs
- Different semantic classes of verbs that show this behavior
- VerbNet has many such classifications and alternations

- Difficult to formally define many roles like this (e.g. intermediary and enabling instruments etc.)

- Need for generalized semantic roles

Alternatives to Thematic Roles

- Less roles: generalized semantic roles, defined as prototypes

- Proposition Bank
  - PROTO-AGENT
  - PROTO-PATIENT

- More roles: Roles specific to a group of predicates (verbs)
  - FrameNet

- For more on Proto-roles, refer to the paper "Semantic Proto-Roles" from TACL 2015.

PropBank and FrameNet

- **PropBank**
  - Uses protoroles: Agent and Patient
    - Agent-like properties: volitionally involved in the event, causing an event or a change of state in another participant, being sentient or intentionally involved
    - Patient-like properties: undergoing change of state, causally affected by another participant, stationary relative to other participants, …
  - and verb-specific semantic roles.
- **FrameNet**
  - Uses semantic roles that are specific to a general semantic idea called a frame.

Proposition Bank (PropBank)

- Arg0: Proto-Agent
  - Volitional involvement (direct/major involvement) in event or state
  - Causes an event or change of state in another participant
  - Movement (relative to position of another participant)

- Arg1: Proto-Patient
  - Undergoes change of state
  - Causally affected by another participant
  - Stationary relative to movement of another participant

- Arg2: benefactive, instrument, attribute, or end state

- Arg3 (start point, instrument, attribute), Arg4 (end point) are less frequent

PropBank Examples

(19.11) **agree.01**

Arg0: Agreer
Arg1: Proposition
Arg2: Other entity agreeing

Ex1: [$_{\text{Arg0}}$ The group] *agreed* [$_{\text{Arg1}}$ it wouldn't make an offer].
Ex2: [$_{\text{ArgM-TMP}}$ Usually] [$_{\text{Arg0}}$ John] *agrees* [$_{\text{Arg2}}$ with Mary] [$_{\text{Arg1}}$ on everything].

(19.12) **fall.01**

Arg1: Logical subject, patient, thing falling
Arg2: Extent, amount fallen
Arg3: start point
Arg4: end point, end state of arg1
Ex1: [$_{\text{Arg1}}$ Sales] *fell* [$_{\text{Arg4}}$ to \$25 million] [$_{\text{Arg3}}$ from \$27 million].
Ex2: [$_{\text{Arg1}}$ The average junk bond] *fell* [$_{\text{Arg2}}$ by 4.2%].

PropBank Examples

Causes an event or change of state in another participant

(19.13) **increase.01** “go up incrementally”
Arg0: causer of increase
Arg1: thing increasing
Arg2: amount increased by, EXT, or MNR
Arg3: start point
Arg4: end point

Experienced the change

(19.14) [$_{\text{Arg0}}$ Big Fruit Co.] increased [$_{\text{Arg1}}$ the price of bananas].
(19.15) [$_{\text{Arg1}}$ The price of bananas] was increased again [$_{\text{Arg0}}$ by Big Fruit Co.]
(19.16) [$_{\text{Arg1}}$ The price of bananas] increased [$_{\text{Arg2}}$ 5%].

No Arg0

Modifiers of Predicates: Arg-M

ArgM-TMP	when?	yesterday evening, now
LOC	where?	at the museum, in San Francisco
DIR	where to/from?	down, to Bangkok
MNR	how?	clearly, with much enthusiasm
PRP/CAU	why?	because ... , in response to the ruling
REC		themselves, each other

- Relatively stable across predicates, so are not listed with each frame file.
- Data labeled with these modifiers can be helpful in training systems to detect temporal, location, or directional modification across predicates.

FrameNet

(19.17) $[_{\text{Arg1}}$ The price of bananas$]$ increased $[_{\text{Arg2}}$ 5%$]$.

(19.18) $[_{\text{Arg1}}$ The price of bananas$]$ rose $[_{\text{Arg2}}$ 5%$]$.

(19.19) There has been a $[_{\text{Arg2}}$ 5%$]$ rise $[_{\text{Arg1}}$ in the price of bananas$]$.

- What about different verbs that mean the same thing?
- Consider the following set of words:
  - Reservation, flight, seat, book, ticket, fare, rate, meals, plane, price
- The words are used in a common-sense background information/context
- This background knowledge is referred to as frame

VERBS:	dwindle	move	soar	escalation	shift
advance	edge	mushroom	swell	explosion	tumble
climb	explode	plummet	swing	fall	
decline	fall	reach	triple	fluctuation	**ADVERBS:**
decrease	fluctuate	rise	tumble	gain	increasingly
diminish	gain	rocket		growth	
dip	grow	shift	**NOUNS:**	hike	
double	increase	skyrocket	decline	increase	
drop	jump	slide	decrease	rise	

coherent chunk of common-sense background information

FrameNet

Frame: Hit_target
(hit, pick off, shoot)

Core	Non-Core
Agent	Means
Target	Place
Instrument	Purpose
Manner	Subregion
	Time

Lexical units (LUs):
Words that evoke the frame
(usually verbs)

Frame elements (FEs):
The involved semantic roles

[$_{\text{Agent}}$ *Kristina*] ***hit*** [$_{\text{Target}}$ *Scott*] [$_{\text{Instrument}}$ *with a baseball*] [$_{\text{Time}}$ *yesterday*].

FrameNet: Core and Non-core Roles

Core Roles	
ATTRIBUTE	The ATTRIBUTE is a scalar property that the ITEM possesses.
DIFFERENCE	The distance by which an ITEM changes its position on the scale.
FINAL_STATE	A description that presents the ITEM's state after the change in the ATTRIBUTE's value as an independent predication.
FINAL_VALUE	The position on the scale where the ITEM ends up.
INITIAL_STATE	A description that presents the ITEM's state before the change in the ATTRIBUTE's value as an independent predication.
INITIAL_VALUE	The initial position on the scale from which the ITEM moves away.
ITEM	The entity that has a position on the scale.
VALUE_RANGE	A portion of the scale, typically identified by its end points, along which the values of the ATTRIBUTE fluctuate.
Some Non-Core Roles	
DURATION	The length of time over which the change takes place.
SPEED	The rate of change of the VALUE.
GROUP	The GROUP in which an ITEM changes the value of an ATTRIBUTE in a specified way.

FrameNet allows frames to inherit from each other, or representing relations between frames

Back to Semantic Roles

- Semantic roles are useful for various tasks.
- Helps in making simple inferences
  - that can not be made from the lexical string of words, or even from the parse tree
- **Question Answering**
  - “Who” questions usually use Agents
  - “What” question usually use Patients
  - “How” and “with what” questions usually use Instruments
  - “To whom” questions usually use Destinations
  - “Where” questions frequently use Sources and Destinations.
  - “For whom” questions usually use Beneficiaries
- **Machine Translation Generation**
  - Semantic roles are usually expressed using particular, distinct syntactic constructions in different languages.

Semantic Role Labeling (SRL) task

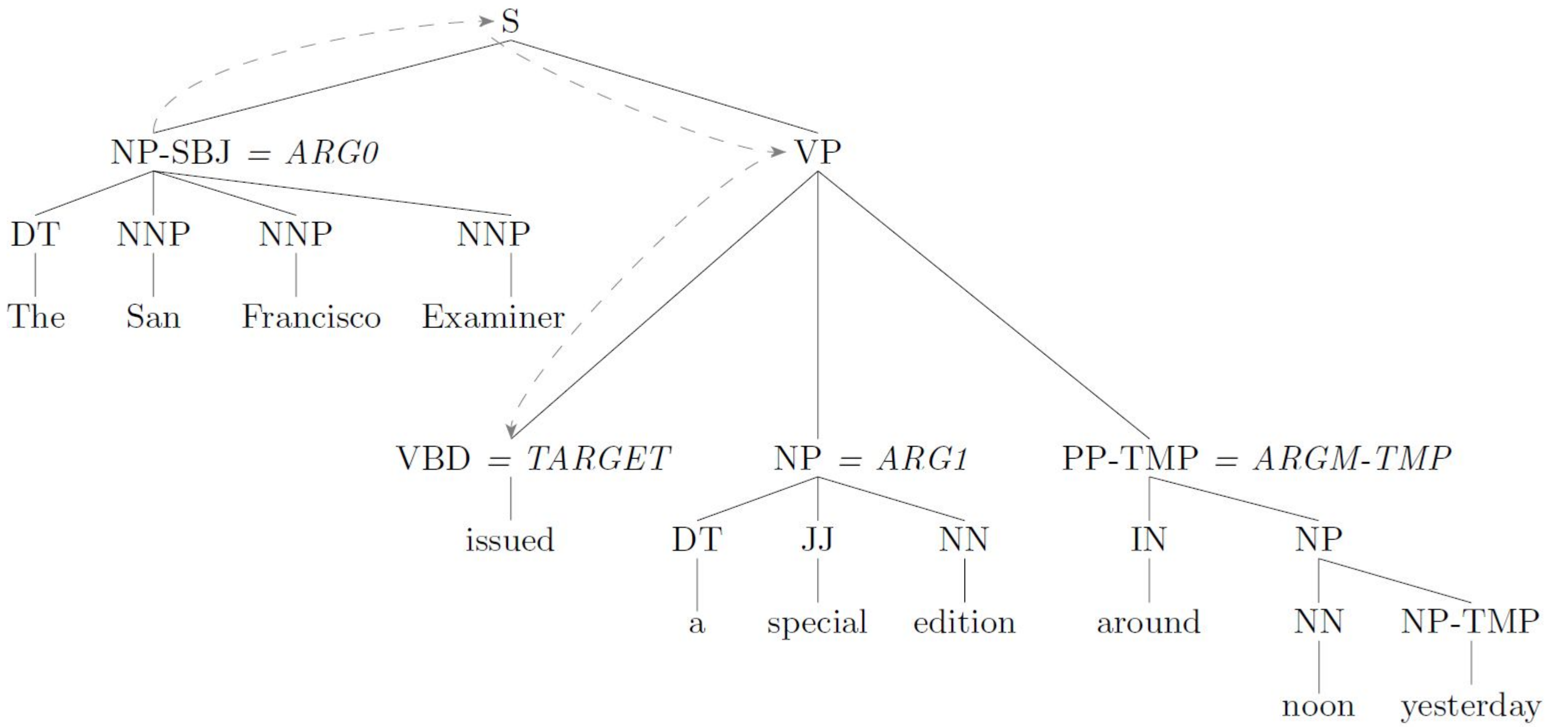
- The task of finding the semantic roles of each argument of each **predicate** in a sentence

- Labels are different for PropBank and FrameNet

[You]	can't	[blame]	[the program]	[for being unable to identify it]
COGNIZER		TARGET	EVALUEE	REASON

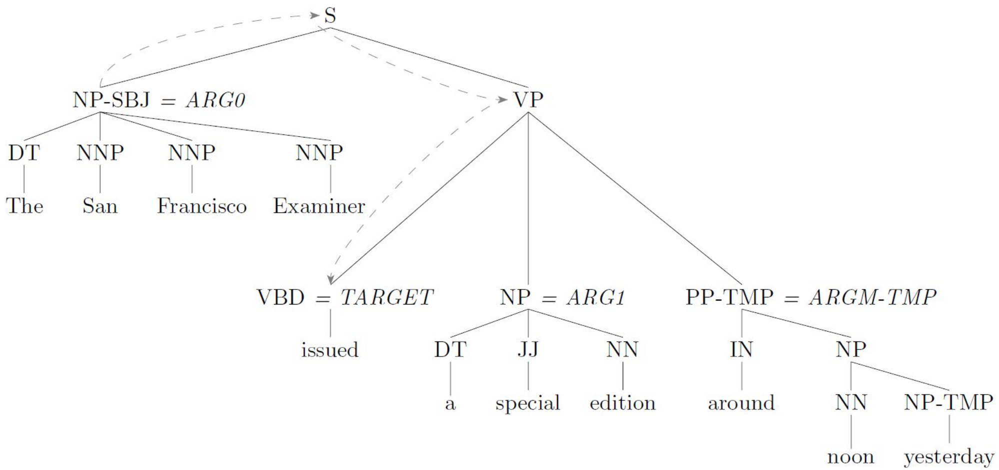
[The San Francisco Examiner]	issued	[a special edition]	[yesterday]
ARG0	TARGET	ARG1	ARGM-TMP

An example

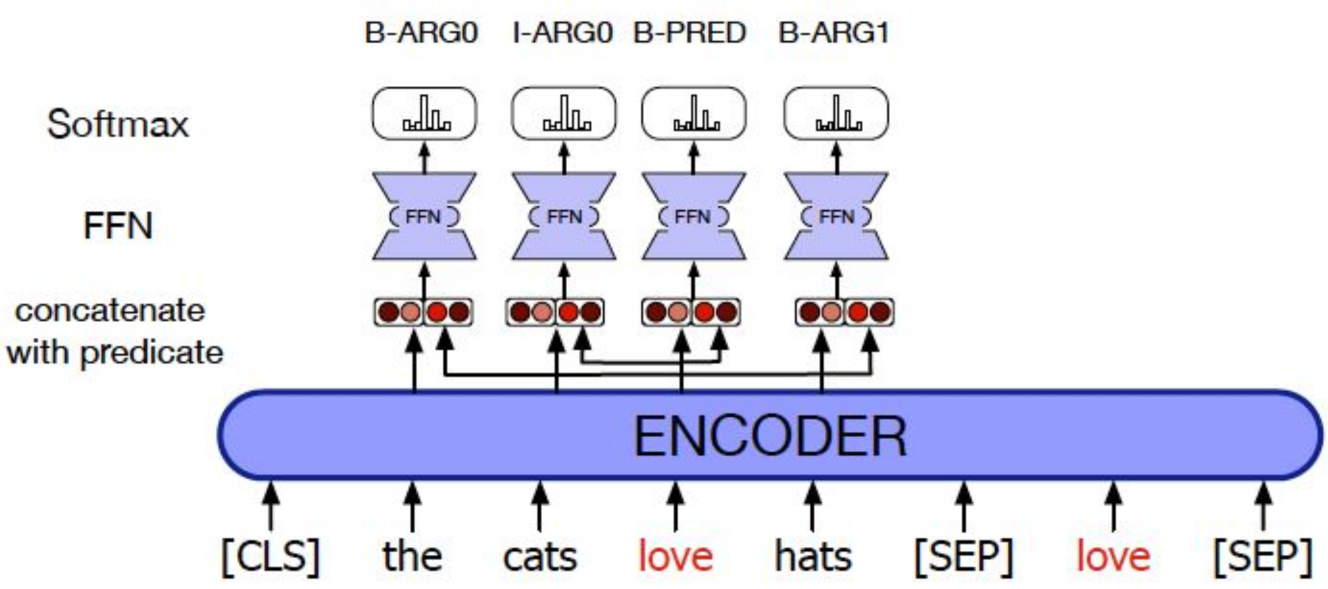

S
NP-SBJ = ARG0
VP
DT
NNP
NNP
NNP
The
San
Francisco
Examiner
VBD = TARGET
NP = ARG1
PP-TMP = ARGM-TMP
issued
DT
JJ
NN
IN
NP
a
special
edition
around
NN
NP-TMP
noon
yesterday

Feature Based Algorithm for SRL

- **Headword of constituent:** Examiner
- **Headword POS:** NNP
- **Voice of the clause:** Active
- **Subcategorization of pred:** VP --> VBD NP PP
- **Named Entity type of constit:** ORG
- **First and last words of constit:** The, Examiner
- **Linear position of clause wrt predicate:** Before
- **Path:** NP(u)S(d)VP(d)VBD

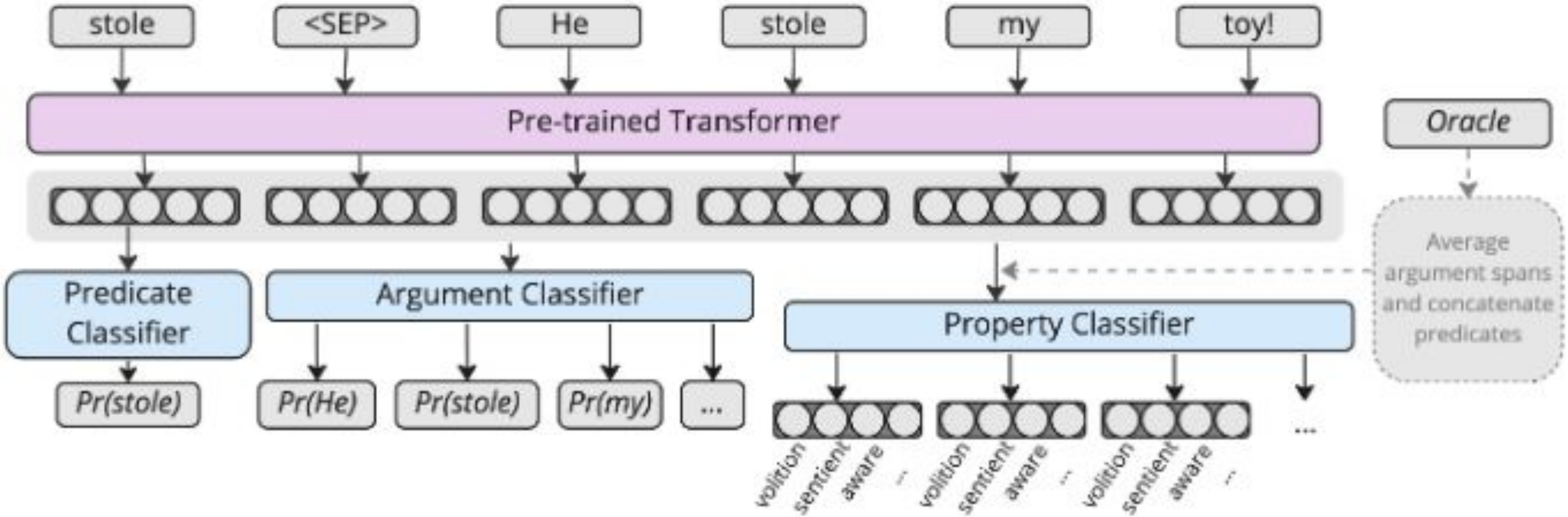


Neural Algorithm for SRL



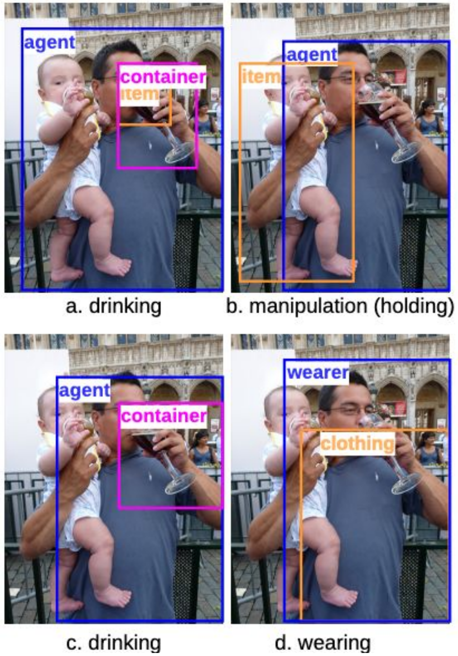
A simple neural approach for semantic role labeling

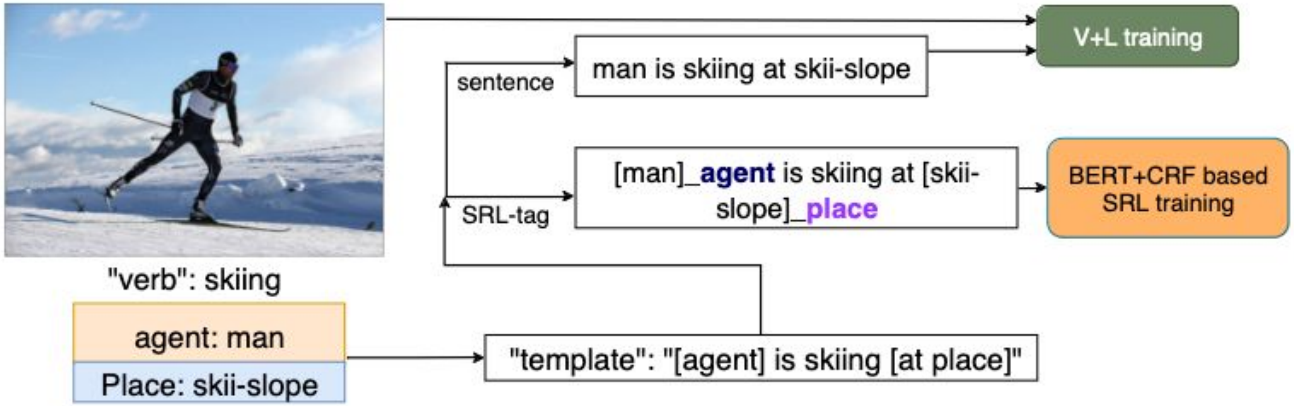
Recent work: Semantic Proto-Role Labeling



Joint End-to-End Semantic Proto-role Labeling. ACL 2023

Recent work: Visual SRL





[Joint End-to-End Semantic Proto-role LabelinCRAPES:Cross-modal Annotation Projection for Visual Semantic Role Labeling.](#) 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)

Natural Language to Arithmetic Expression

- Objective
  - Interpret natural language arithmetic expression
- Sample Data

Natural Language Expression	Semantic Representation	Denotation
One plus one	(+ , 1 , 1)	2
Minus three minus two	(- , (- , 3) , 2)	-5
Three plus three minus two	(- , (+ , 3 , 3), 2)	4
Two times two plus three	(+ , (*, 2, 2), 3)	7

Natural Language to logical-form

- Objective
  - Map mapping natural language sentences to lambda–calculus
- Sample data

a) What states border Texas
$\lambda x.state(x) \wedge borders(x, texas)$

b) What is the largest state
$\arg\max(\lambda x.state(x), \lambda x.size(x))$

c) What states border the state that borders the most states
$\lambda x.state(x) \wedge borders(x, \arg\max(\lambda y.state(y),$
$\lambda y.count(\lambda z.state(z) \wedge borders(y, z))))$

Travel Queries

- Objective
  - Interpret travel queries
    - Applied in intelligent assistants such as Apple's Siri, Google Now, MS Cortana, and Amazon Echo
- Dataset
  - AOL Search query
- Sample data

Natural Language Query	Semantic Representation
Directions from Washington to Canada	{**domain**: ‘travel’, **type**: ‘directions’, **origin**: {**id**: 4140963, **name**: 'Washington, DC, US'}, **destination**: {**id**: 6251999, **name**:'Canada'}}
Cost of an airfare from Newark to Charleston	{**domain**: 'travel', **type**: 'cost', **mode**: 'air', **origin**: {**id**: 5101798, **name**: 'Newark, NJ, US'}, **destination**: {**id**: 4574324, **name**: 'Charleston, SC, US'}}

Natural Language to SQL

- Objective
  - Convert natural language to SQL query
- Dataset
  - WikiSQL , Spider , CoSQL , SParC
- Sample data (WikiSQL)

Table

Player	No.	Nationality	Position	Years in Toronto	School/Club Team
Antonio Lang	21	United States	Guard-Forward	1999-2000	Duke
Voshon Lenard	2	United States	Guard	2002-03	Minnesota
Martin Lewis	32, 44	United States	Guard-Forward	1996-97	Butler CC (KS)
Brad Lohaus	33	United States	Forward-Center	1996	Iowa
Art Long	42	United States	Forward-Center	2002-03	Cincinnati

Question:

Who is the player that wears number 42?

SQL:

```
SELECT player
WHERE no. = 42
```

Result:

Art Long

Text-to-SQL Model (WikiSQL task)

- Features of WikiSQL
  - Format of the queries are mostly SELECT <col> from TABLE WHERE<col> <op> <value>
  - WHERE clause contains only a single constraint
  - Each query is associated with a single table
  - No complex queries (No keywords like JOIN, WITH, GROUP BY, HAVING etc.)
- How to approach towards solution?
  - Can view the task as slot filling (**select <col>, where <col>, <op> and <value>**)
- Basic Idea
  - Encode natural language question and database schema (using Sequence-Sequence or Transformer based architecture)
  - Fill the slots using this encoded information
- Reference
  - SQLNET
  - IRNET