

SD-TSIA214: Structured prediction in natural language processing

Maria Boritchev

March 22, 2024

Télécom Paris, Institut Polytechnique de Paris

Introduction

What is structured prediction?

Tasks: predict structured outputs \neq scalar value/categorical quantity

What is structured prediction?

Tasks: predict structured outputs \neq scalar value/categorical quantity

What NLP tasks do you know?

What is structured prediction?

Tasks: predict structured outputs \neq scalar value/categorical quantity

What NLP tasks do you know?

Not structured prediction sentiment analysis, text classification,
chatbots, autocorrection, speech recognition

Structured prediction tagging, parsing, coreference resolution, text
summarization

Depends? machine translation, natural language understanding,
natural language inference

Part-of-Speech (POS) tagging

Named Entity Recognition (NER)

Dependency parsing

Part-of-Speech (POS) tagging

Example:

Charlie likes green tea

Part-of-Speech (POS) tagging

Example:

Charlie	likes	green	tea
NNP	VBZ	JJ	NN

Part-of-Speech (POS) tagging

Example:

Charlie	likes	green	tea
NNP	VBZ	JJ	NN

What do the tags stand for?

Part-of-Speech (POS) tagging

Example:

Charlie	likes	green	tea
NNP	VBZ	JJ	NN

What do the tags stand for?

NNP	Proper noun
VBZ	Verb
JJ	Adjective
NN	Noun

Why is POS tagging interesting and useful?

Computational linguistics tasks:

- **Linguistic change:** creation of new words, meaning shift
- **Linguistic variation:** regional, social, medical
- **Comparison and control:** use POS to measure meaning similarity

Why is POS tagging interesting and useful?

Computational linguistics tasks:

- **Linguistic change:** creation of new words, meaning shift
- **Linguistic variation:** regional, social, medical
- **Comparison and control:** use POS to measure meaning similarity

NLP tasks:

- Syntactic parsing
- **Machine translation:** word order (SOV/SVO/VSO etc.)
- **Sentiment analysis:** adjectives vs. other POS
- **Text-to-speech:** pronunciation disambiguation

POS tags – Universal Dependencies (UD)

ADJ	adjective
ADP	adposition
ADV	adverb
AUX	auxiliary
CCONJ	coordinating conjunction
DET	determiner
INTJ	interjection
NOUN	noun
NUM	numeral
PART	particle
PRON	pronoun
PROPN	proper noun
PUNCT	punctuation
SCONJ	subordinating conjunction
SYM	symbol
VERB	verb
X	other

Exercise/example: POS tags

ADJ	adjective
AUX	auxiliary
DET	determiner
NOUN	noun
PART	particle
PRON	pronoun
PROPN	proper noun
VERB	verb

Find tagging errors in the following sentence:

Does/AUX this/PRON flight/NOUN serve/VERB dinner/NOUN

Is POS tagging a difficult task in English?

Ambiguity: $\sim 15\%$ of words are ambiguous

Is POS tagging a difficult task in English?

Ambiguity: ~ 15% of words are ambiguous

Example: “back”

- Earnings growth took a back/ADJ seat
- A small building in the back/NOUN
- A clear majority of senators back/VERB the bill
- Enable the country to buy back/PART debt
- I was twenty-one back/ADV then

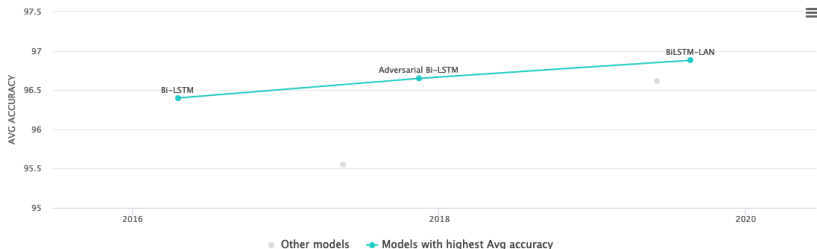
SOTA performance of POS tagging in English

Part-Of-Speech Tagging on UD

Leaderboard

Dataset

View Avg accuracy by Date



Filter: LSTM untagged

Edit Leaderboard

Rank	Model	Avg accuracy	Paper	Code	Result	Year	Tags
1	BiLSTM-LAN	96.88	Hierarchically-Refined Label Attention Network for Sequence Labeling	GitHub	Result	2019	LSTM

Example: “The horse raced past the barn fell.”

Exemple: « L’oiseau vola tout en douceur au bord de la fenêtre le diamant qui y était déposé. »

Example:

- The/**DET** horse/**NOUN** raced/**VERB** past/**NOUN** the/**DET** barn/**NOUN** fell/**VERB**
- **Explanation:** “the horse **which was** raced past the barn fell”

Example: « L’oiseau vola tout en douceur au bord de la fenêtre le diamant qui y était déposé »

Sources of information for POS tagging

Example: “Janet will back the bill”

Sources of information for POS tagging

Example: “Janet will back the bill”

- “will”: AUX/NOUN/VERB?
- “bill”: NOUN/VERB?

Sources of information for POS tagging

Example: “Janet will back the bill”

- “will”: **AUX/NOUN/VERB?**
- “bill”: **NOUN/VERB?**

Some contextual clues:

- Prior probabilities of word/tag:
 - “will” is usually an **AUX**
- Neighboring words:
 - “the” means the next word is probably not a **VERB**
- Morphology and wordshape:

Prefixes: un- → **ADJ** **ex:** unable

Suffixes: -ly → **ADJ** **ex:** importantly

Capitalization: CAP → **PROPN** **ex:** Janet

Exercise/example: simple grammar

Abbreviation	Name	Example
N	noun	"tea"
PN	proper noun	"Charlie"
TV	transitive verb	"likes"
Adj	adjective	"green"
D	determiner	"the"

Abbreviation	Name	Example
NP \rightarrow D N	noun phrase	"the unicorn"
VP \rightarrow TV NP	verb phrase	"likes tea"
S \rightarrow NP VP	sentence	"Charlie likes tea"

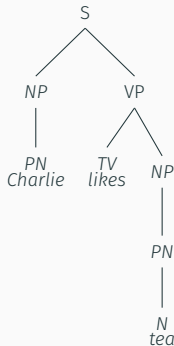
Exercise/example: “Charlie likes tea” – A minimal grammar

Lexicon: $\left\{ \begin{array}{lll} PN & \rightarrow & \text{Charlie} \\ N & \rightarrow & \text{tea} \\ TV & \rightarrow & \text{likes} \end{array} \right.$

Grammar: $\left\{ \begin{array}{lll} S & \rightarrow & NP\ VP \\ VP & \rightarrow & TV\ NP \\ NP & \rightarrow & PN \\ PN & \rightarrow & N \end{array} \right.$

Exercise/example: “Charlie likes tea” – A minimal grammar

Lexicon: $\left\{ \begin{array}{ll} PN & \rightarrow \text{Charlie} \\ N & \rightarrow \text{tea} \\ TV & \rightarrow \text{likes} \end{array} \right.$ Grammar: $\left\{ \begin{array}{ll} S & \rightarrow NP \ VP \\ VP & \rightarrow TV \ NP \\ NP & \rightarrow PN \\ PN & \rightarrow N \end{array} \right.$



A syntactic tree!

Exercise: “The unicorn drinks tea”

Abbreviation	Name	Example
N	noun	“tea”
PN	proper noun	“Charlie”
TV	transitive verb	“likes”
D	determiner	“the”

$$\text{Grammar: } \left\{ \begin{array}{lcl} S & \rightarrow & NP \ VP \\ VP & \rightarrow & TV \ NP \\ NP & \rightarrow & PN \\ NP & \rightarrow & DN \\ PN & \rightarrow & N \end{array} \right.$$

Using the lexicon and the grammar rules, draw the syntactic tree of the sentence *“The unicorn drinks tea”*.

Supervised Machine Learning Algorithms:

- Hidden Markov Models (HMM)
- Conditional Random Fields (CRF)
- Maximum Entropy Markov Models (MEMM)
- Neural sequence models (RNNs or Transformers)
- Finetuned LLMs (ex: BERT)



All of these required a human-labeled training set!

Named Entity Recognition (NER)

What are Named Entities?

A **named entity** (NE) is anything that can be referred to with a proper name. Most common NE are of one of the four types:

- **People:** “Marie Curie”
- **Locations:** “Paris”
- **Organizations:** “Institut Polytechnique de Paris”
- **Geo-Political Entities:** “European Union”

The term is now extended: multi-word expressions, dates, times, prices.

Exercise/example: NE tagging/NER

Task:

- (1) find spans of text constituting the NE;
- (2) tag the type (PEO, LOC, ORG, GPE, AMOUNT) of this NE.

Example:

The STA Plaza (The Plaza or Spokane Transit Authority Plaza), is a transit center located in Downtown Spokane, Washington. It is the main hub of customer service and transit operations for the Spokane Transit Authority (STA), with 31 out of its 52 bus routes connecting with The Plaza.

Exercise/example: NE tagging/NER

Task:

- (1) find spans of text constituting the NE;
- (2) tag the type (PEO, LOC, ORG, GPE, AMOUNT) of this NE.

(1):

The [STA Plaza] ([The Plaza] or [Spokane Transit Authority Plaza]), is a transit center located in [Downtown Spokane, Washington]. It is the main hub of customer service and transit operations for the [Spokane Transit Authority (STA)], with [31] out of its [52] bus routes connecting with [The Plaza].

Exercise/example: NE tagging/NER

Task:

- (1) find spans of text constituting the NE;
- (2) tag the type (PEO, LOC, ORG, GPE, AMOUNT) of this NE.

(2):

The *STA Plaza/LOC* (*The Plaza/LOC* or *Spokane Transit Authority Plaza/LOC*), is a transit center located in *Downtown Spokane, Washington/LOC*. It is the main hub of customer service and transit operations for the *Spokane Transit Authority (STA)/ORG*, with *31/AMOUNT* out of its *52/AMOUNT* bus routes connecting with *The Plaza/LOC*.

Exercise: NER vs proper nouns (1)

In the following list, identify the NE:

Carcassonne, Napoléon, Vivendi, EDF, Dubonnet, United Nations, SNCF, French revolution, Clio, prime minister, French, table

Exercise: NER vs proper nouns (2)

How many NE are there in the following sentences?

1. This idea was put forward by some philosophers in the 18th and 19th centuries such as Mary Wollstonecraft and John Stuart Mill.
2. These women were called Suffragettes because they were fighting for Universal suffrage which means everybody is able to vote.

Exercise: NER - heuristics

Propose two examples of polysemic NE.

Propose characteristic elements of EN for people, places and organizations:

1. At the level of the NEs themselves.
2. According to the context in which they appear.

Why is NER interesting and useful?

Examples in NLP applications:

- **Sentiment analysis:** consumer's sentiment towards a particular company or person?
- **Question answering:** answer questions about an entity?
- **Information extraction:** extracting facts about entities from text

Is NER a difficult task in English?

Segmentation: where does a NE start/end?

Type ambiguity: can a NE be of different types depending on the context?

Examples:

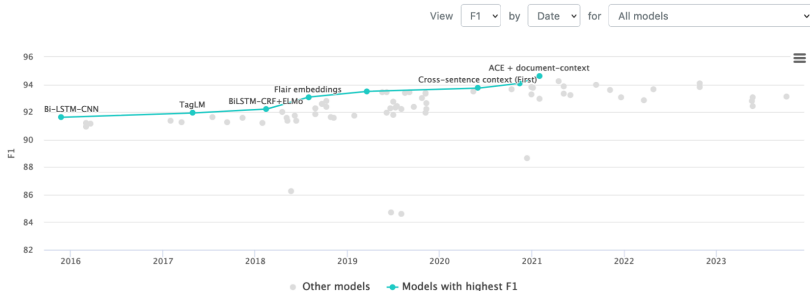
- *Washington/PER was born into slavery on the farm of James Burroughs.*
- *Washington/ORG went up 2 games to 1 in the four-game series.*
- *Blair arrived in Washington/LOC for what may well be his last state visit.*

SOTA performance of NER in English

Named Entity Recognition (NER) on CoNLL 2003 (English)

Leaderboard

Dataset



Filter:

LSTM

Transformer

MRC-based

knowledge distillation

GCN

tagged

Edit Leaderboard

Rank	Model	Extra F1 ↑	Training Data	Paper	Code	Result	Year	Tags
------	-------	------------	---------------	-------	------	--------	------	------

1 ACE + document-context

94.6

×

Automated Concatenation of Embeddings for
Structured Prediction



2021

LSTM

Transformer

Beginning-inside-outside (BIO) tagging: merging (1) and (2) NER tasks into one tagging task.

Example:

Charlie is going to Los Angeles

Beginning-inside-outside (BIO) tagging: merging (1) and (2) NER tasks into one tagging task.

Example:

Charlie	is	going	to	Los	Angeles
I-PER	O	O	O	B-LOC	I-LOC

Supervised Machine Learning Algorithms:

- Hidden Markov Models (HMM)
- Conditional Random Fields (CRF)
- Maximum Entropy Markov Models (MEMM)
- Neural sequence models (RNNs or Transformers)
- Finetuned LLMs (ex: BERT)



All of these required a human-labeled training set!

Coreference resolution (CR) is the task of finding all linguistic expressions in a given text that refer to the same entity.

Example:

Sam saw a white star twinkle for a while. The beauty of it smote his heart, as he looked up out of the forsaken land, and hope returned to him.

Coreference resolution (CR) is the task of finding all linguistic expressions in a given text that refer to the same entity.

Example:

Sam saw a white star twinkle for a while. The beauty of it smote his heart, as he looked up out of the forsaken land, and hope returned to him.

A **sequence labeler** is a model that assigns a label to each unit in a sequence, mapping a sequence of observations to a sequence of labels of the same length.

Sequence labeling algorithms

A **sequence labeler** is a model that assigns a label to each unit in a sequence, mapping a sequence of observations to a sequence of labels of the same length.

Examples:

Charlie	likes	green	tea
NNP	VBZ	JJ	NN

Charlie	is	going	to	Los	Angeles
I-PER	O	O	O	B-LOC	I-LOC

See jupyter notebook tutorial.

Goal: add information about the context in which a word appears.

- What is the previous word? **AND** the next word?
- What is the syntactic shape of the sentence parts before and after the word?
- Are there unknown words?

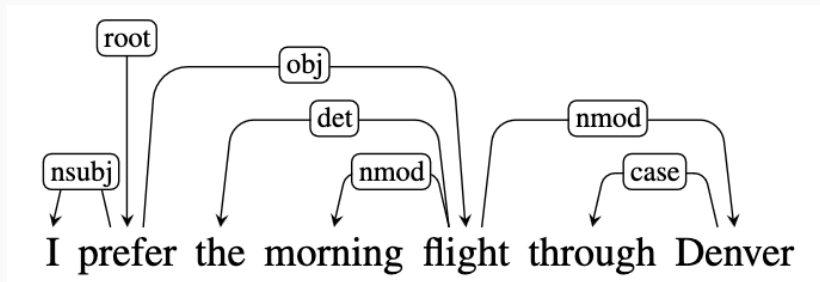
HMM vs Conditional Random Fields: (CRF) and **Maximum Entropy Markov Models** (MEMM) allow integration of rich features for better accuracy.

Cost: much slower training!

Dependency parsing

Dependency parsing

In **dependency parsing**, the syntactic structure of a sentence is described in terms of directed binary grammatical relations between the words:



The arcs go from **heads** to **dependents**; the parse is a **typed dependency structure**.

→ Dependency structure shows which words depend on (modify, attach to, or are arguments of) which other words.

Why is dependency parsing interesting and useful?

Computational linguistics tasks:

- **Human communication:** compositionality for complex sentences and meanings
- **Linguistic variation:** multilingual, regional, social, medical

Why is dependency parsing interesting and useful?

Computational linguistics tasks:

- **Human communication:** compositionality for complex sentences and meanings
- **Linguistic variation:** multilingual, regional, social, medical

NLP tasks:

- Syntactic parsing for **semantic parsing**
- **Machine translation:** word order (SOV/SVO/VSO etc.)
- **Sentiment analysis:** adjectives vs. other POS
- **Text-to-speech:** pronunciation disambiguation

Phrase attachment ambiguities

Examples:

- Scientists count whales from space
- Teacher strikes idle kids
- Enraged cow injures farmer with axe
- Miners refuse to work after death

<https://www.departments.bucknell.edu/linguistics/synhead.html>

A **dependency structure** can be represented as a directed graph

$G = (V, A)$:

- a set of vertices V (\sim set of words in a given sentence + punctuation)
- a set of ordered pairs of vertices A (arcs)

A **dependency structure** can be represented as a directed graph $G = (V, A)$:

- a set of vertices V (\sim set of words in a given sentence + punctuation)
- a set of ordered pairs of vertices A (arcs)

Different grammatical theories or formalisms may place further constraints on these dependency structures:

- the structures must be connected
- have a designated root node
- be acyclic or planar.

Dependency formalism

A **dependency tree** is a directed graph that satisfies the following constraints:

1. There is a single designated root node that has no incoming arcs.
2. With the exception of the root node, each vertex has exactly one incoming arc.
3. There is a unique path from the root node to each vertex in V .

Thus:

- each word has a single head;
- the dependency structure is connected;
- there is a single root node from which one can follow a unique directed path to each of the words in the sentence.

Dependency relations

Clausal Argument Relations	Description
NSUBJ	Nominal subject
OBJ	Direct object
IOBJ	Indirect object
CCOMP	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

Figure 18.2 Some of the Universal Dependency relations (de Marneffe et al., 2021).

Dependency relations examples

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

Figure 18.3 Examples of some Universal Dependency relations.

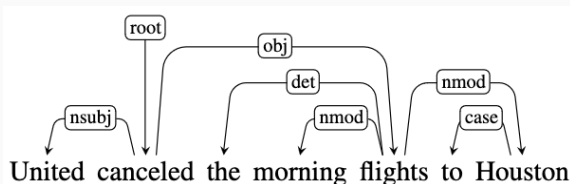
Exercise/example

Relation	Description & example (with head and dependent)
ROOT	Root of the tree, head of the entire structure
NSUBJ	Nominal subject, ex: United canceled the flight.
OBJ	Direct object, ex: We booked her the first flight to Miami.
NMOD	Nominal modifier, ex: We took the morning flight .
DET	Determiner, ex: The flight was canceled.
CASE	Prepositions, postpositions, other case markers, ex: Book the flight through Houston .

Exercise: What is the dependency parse of the sentence
“United canceled the morning flights to Houston”?

Exercise/example

Relation	Description & example (with head and dependent)
ROOT	Root of the tree, head of the entire structure
NSUBJ	Nominal subject, ex: United canceled the flight.
OBJ	Direct object, ex: We booked her the first flight to Miami.
NMOD	Nominal modifier, ex: We took the morning flight .
DET	Determiner, ex: The flight was canceled.
CASE	Prepositions, postpositions, other case markers, ex: Book the flight through Houston .



Universal Dependencies project - more than 100 languages

Current UD Languages

Information about language families (and genera for families with multiple branches) is mostly taken from [WALS Online](https://wals.info/) (IE = Indo-European).

▶		Abaza	1	<1K	☞	Northwest Caucasian
▶		Afrikaans	1	49K	🗺️🇺🇸	IE, Germanic
▶		Akkadian	2	25K	📖🇺🇸	Afro-Asiatic, Semitic
▶		Akuntsu	1	1K	📖🇺🇸	Tupian, Tupari
▶		Albanian	1	<1K	🇲🇰	IE, Albanian
▶		Amharic	1	10K	🗺️🇺🇸🇲🇰🇺🇸	Afro-Asiatic, Semitic
▶		Ancient Greek	2	416K	🗺️🇺🇸🇲🇰	IE, Greek
▶		Ancient Hebrew	1	39K	🗺️	Afro-Asiatic, Semitic
▶		Apurina	1	<1K	📖🇺🇸	Arawakan
▶		Arabic	3	1,042K	📖🇲🇰	Afro-Asiatic, Semitic
▶		Armenian	2	94K	📖🗺️🇺🇸🇲🇰🇺🇸🇲🇰🇺🇸	IE, Armenian
▶		Assyrian	1	<1K	📖🇺🇸	Afro-Asiatic, Semitic
▶		Bambara	1	13K	📖🇺🇸	Mande
▶		Basque	1	121K	📖	Basque
▶		Beja	1	<1K	☞	Afro-Asiatic, Cushitic
▶		Belarusian	1	305K	🗺️🇺🇸🇲🇰🇺🇸🇲🇰	IE, Slavic
▶		Bengali	1	<1K	🗺️	IE, Indic
▶		Bhojpuri	1	6K	📖🇺🇸	IE, Indic
▶		Bororo	1	<1K	🗺️	Bororoan
▶		Breton	1	10K	🗺️🇺🇸🇲🇰🇺🇸🇲🇰	IE, Celtic
▶		Bulgarian	1	156K	🗺️🇺🇸	IE, Slavic
▶		Buryat	1	10K	🗺️🇺🇸	Mongolic
▶		Cantonese	1	13K	☞	Sino-Tibetan
▶		Catalan	1	553K	📖	IE, Romance
▶		Cebuano	1	1K	🗺️	Austronesian, Central Philippine
▶		Chinese	6	287K	🗺️🇺🇸🇲🇰🇺🇸	Sino-Tibetan
▶		Chukchi	1	6K	☞	Chukotko-Kamchatkan
▶		Classical Chinese	1	433K	🗺️🇺🇸	Sino-Tibetan
▶		Coptic	1	55K	🗺️🇺🇸	Afro-Asiatic, Egyptian

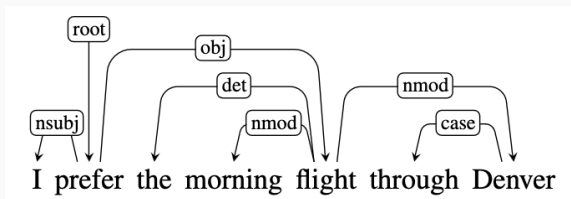
Treebanks play a critical role in the development and evaluation of dependency parsers:

- used for training parsers;
- act as the gold labels for evaluating parsers;
- provide useful information for corpus linguistics studies.

Dependency treebanks are created by having **human annotators** directly create dependency structures for a given corpus, or by hand-correcting the output of an automatic parser.

Sources of information for dependency parsing

Example:



- **Bilexical affinities:** the dependency [flight → morning] is plausible
- **Dependency distance:** Most dependencies are between nearby words
- **Intervening material:** Dependencies rarely span intervening words (ex: like, with, plus, including) or punctuation
- **Valency of heads:** How many dependents on which side are usual for a head?

Transition-based dependency parsing

Three components:

- (1) a **stack**, on which the parse is built
- (2) a **buffer**, containing the tokens to be parsed
- (3) a **parser** which takes actions on the parse via a predictor: an **oracle**

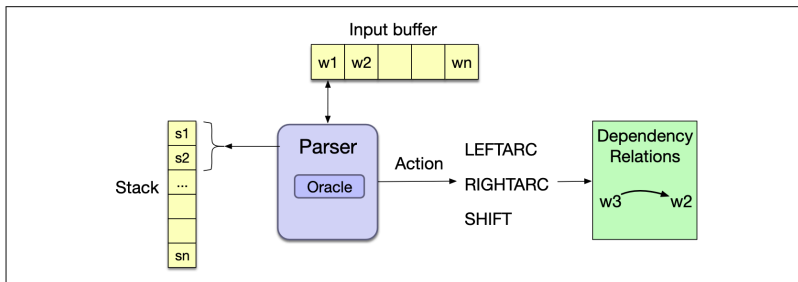


Figure 18.4 Basic transition-based parser. The parser examines the top two elements of the stack and selects an action by consulting an oracle that examines the current configuration.

Transition-based dependency parsing

The **parser**: walks through the sentence left-to-right, shifting items from the **buffer** onto the **stack**.

Transition-based dependency parsing

The **parser**: walks through the sentence left-to-right, shifting items from the **buffer** onto the **stack**.

At each time point: the top two elements on the **stack** are examined, the **oracle** makes a decision about what transition to apply to build the parse:

- **LEFTARC**: assign the current word as the head of some previously seen word
- **RIGHTARC**: assign some previously seen word as the head of the current word;
- **SHIFT**: postpone dealing with the current word, storing it for later processing.

Transition-based dependency parsing

The **parser**: walks through the sentence left-to-right, shifting items from the **buffer** onto the **stack**.

At each time point: the top two elements on the **stack** are examined, the **oracle** makes a decision about what transition to apply to build the parse:

- **LEFTARC**: assign the current word as the head of some previously seen word
- **RIGHTARC**: assign some previously seen word as the head of the current word;
- **SHIFT**: postpone dealing with the current word, storing it for later processing.

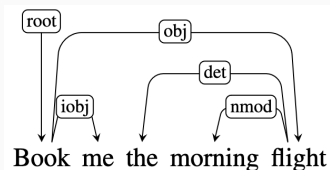
⚠ **LEFTARC** cannot be applied when ROOT is the second element of the stack.

⚠ **LEFTARC** and **RIGHTARC** require two elements to be on the stack to be applied.

Example: transition-based parse trace

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	

Figure 18.6 Trace of a transition-based parse.

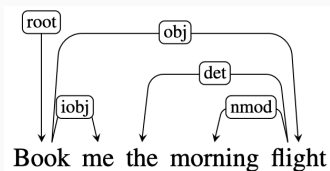


The **oracle** for greedily selecting the appropriate transition is trained by supervised machine learning: **need training data!**

Step	Stack	Word List	Predicted Action
0	[root]	[book, the, flight, through, houston]	SHIFT
1	[root, book]	[the, flight, through, houston]	SHIFT
2	[root, book, the]	[flight, through, houston]	SHIFT
3	[root, book, the, flight]	[through, houston]	LEFTARC
4	[root, book, flight]	[through, houston]	SHIFT
5	[root, book, flight, through]	[houston]	SHIFT
6	[root, book, flight, through, houston]	[]	LEFTARC
7	[root, book, flight, houston]	[]	RIGHTARC
8	[root, book, flight]	[]	RIGHTARC
9	[root, book]	[]	RIGHTARC
10	[root]	[]	Done

Figure 18.7 Generating training items consisting of configuration/predicted action pairs by simulating a parse with a given reference parse.

Dependency parse encoding



A possible encoding for this parse would be:

$$\{(0, 1), (1, 5), (1, 2), (5, 3), (5, 4)\}$$

How is this encoding produced?

How to evaluate parsing?

How to evaluate parsing?

- Exact match (EM): how many sentences are parsed correctly
- → **most sentences are marked wrong**

How to evaluate parsing?

- **Exact match (EM):** how many sentences are parsed correctly
- → **most sentences are marked wrong**
- **Labeled attachment score (LAS):** is a word properly assigned to its head with the correct dependency relation?
- **Unlabeled attachment score (UAS):** is a word properly assigned to its head? (ignoring the dependency relation)
- **Label accuracy score (LS):** what is the percentage of tokens with correct labels? (ignoring where the relations come from)

How to evaluate parsing?

- **Exact match (EM)**: how many sentences are parsed correctly
- → **most sentences are marked wrong**
- **Labeled attachment score (LAS)**: is a word properly assigned to its head with the correct dependency relation?
- **Unlabeled attachment score (UAS)**: is a word properly assigned to its head? (ignoring the dependency relation)
- **Label accuracy score (LS)**: what is the percentage of tokens with correct labels? (ignoring where the relations come from)

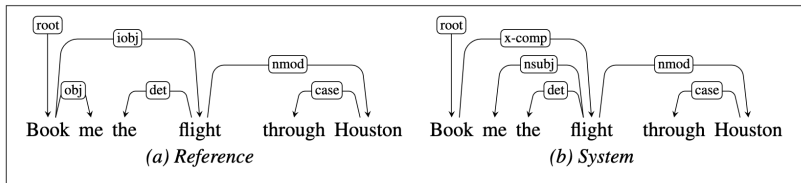


Figure 18.15 Reference and system parses for *Book me the flight through Houston*, resulting in an LAS of 2/3 and an UAS of 5/6.

Conclusion

Structured prediction: prediction of structured outputs, such as

- **Part-of-speech (POS) tagging**
- **Named Entity Recognition (NER)**
- **Coreference resolution**

Historically, human-crafted rule-based grammars were used. Now, prediction algorithms such as **Hidden Markov Models (HMM)**, and more advanced ones such as Conditional Random Fields (CRF) and Maximum Entropy Markov Models (MEMM).

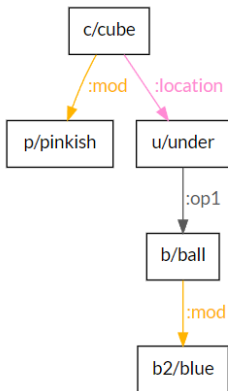
A more advanced example of a structured prediction task is **dependency parsing**, for which both feature-based models and prediction algorithms are needed.

Semantic parsing

AMR parsing [3, 1]:

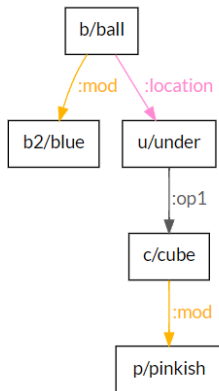
1: A pinkish cube under a blueish ball.

```
(c / cube
 :mod (p / pinkish)
 :location (u / under
           :op1 (b / ball
                 :mod (b2 / blue))))
```



2: A blueish ball under a pinkish cube.

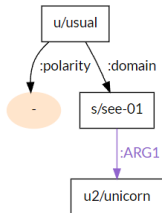
```
(b / ball
 :mod (b2 / blue)
 :location (u / under
           :op1 (c / cube
                 :mod (p / pinkish))))
```



Semantic parsing and semantics

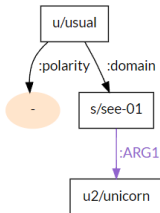
1: It's not usual to see unicorns.

```
(u / usual  
 :polarity -  
 :domain (s / see-01  
 :ARG1 (u2 / unicorn)))
```



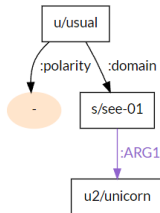
2: It's unusual to see unicorns.

```
(u / usual  
 :polarity -  
 :domain (s / see-01  
 :ARG1 (u2 / unicorn)))
```



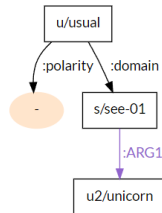
3: It's not unusual to see unicorns.

```
(u / usual  
 :polarity -  
 :domain (s / see-01  
 :ARG1 (u2 / unicorn)))
```



4: It's not not usual to see unicorns.

```
(u / usual  
 :polarity -  
 :domain (s / see-01  
 :ARG1 (u2 / unicorn)))
```



Structured Prediction in NLP - A survey

Chauhan Dev Naman Biyani Nirmal P. Suthar

Prashant Kumar Priyanshu Agarwal

{devgiri, namanb, nirmalps, praskr, priyanag}@iitk.ac.in

Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

Over the last several years, the field of Structured prediction in NLP has had seen huge advancements with sophisticated probabilistic graphical models, energy-based networks, and its combination with deep learning-based approaches. This survey provides a brief of major techniques in structured prediction and its applications in the NLP domains like parsing, sequence labeling, text generation, and sequence to sequence tasks. We also deep-dived into energy-based and attention-based techniques in structured prediction, identified some relevant open issues and gaps in the current state-of-the-art research, and have come up with some detailed ideas for future research in these fields.

ACL 2022 6th Workshop on Structured Prediction for NLP

[Call for Papers](#) [Important Dates](#) [Schedule](#) [Accepted Papers](#) [Organizers](#) [Anti-Harassment Policy](#)

Accepted Papers

Accepted Papers

- *A Joint Learning Approach for Semi-supervised Neural Topic Modeling*. Jeffrey Chiu, Rajat Mittal, Neehal Tumma, Abhishek Sharma and Finale Doshi-Velez
- *SlotGAN: Detecting Mentions in Text via Adversarial Distant Learning*. Daniel Daza, Michael Cochez and Paul Groth
- *TempCaps: A Capsule Network-based Embedding Model for Temporal Knowledge Graph Completion*. Guirong Fu, Zhao Meng, Zhen Han, Zifeng Ding, Yunpu Ma, Matthias Schubert, Volker Tresp and Roger Wattenhofer
- *Joint Entity and Relation Extraction Based on Table Labeling Using Convolutional Neural Networks*. Youmi Ma, Tatsuya Hiraoka and Naoaki Okazaki
- *Multilingual Syntax-aware Language Modeling through Dependency Tree Conversion*. Shunsuke Kando, Hiroshi Noji and Yusuke Miyao
- *Predicting Attention Sparsity in Transformers*. Marcos Vinicius Treviso, António Góis, Patrick Fernandes, Erick Rocha Fonseca, and Andre Martins
- *Neural String Edit Distance*. Jindřich Libovický and Alexander Fraser

<https://www.cs.cornell.edu/courses/cs6741/2015fa/>

https://web.stanford.edu/~jurafsky/slp3/slides/8_POSNER_intro_May_6_2021.pdf

<https://universaldependencies.org/u/pos/>

[https://paperswithcode.com/sota/
part-of-speech-tagging-on-ud](https://paperswithcode.com/sota/part-of-speech-tagging-on-ud)

[https://paperswithcode.com/task/
named-entity-recognition-ner](https://paperswithcode.com/task/named-entity-recognition-ner)

<https://web.stanford.edu/~jurafsky/slp3/8.pdf>

<https://web.stanford.edu/~jurafsky/slp3/18.pdf>

<https://universaldependencies.org/>

<https://web.stanford.edu/class/cs224n/slides/cs224n-2023-lecture04-dep-parsing.pdf>

https://en.wikipedia.org/wiki/K-means_clustering



L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider.
Abstract meaning representation for sembanking.

In Proceedings of the 7th linguistic annotation workshop and interoperability with discourse, pages 178–186, 2013.



C. Dev, N. Biyani, N. P. Suthar, P. Kumar, and P. Agarwal.
Structured Prediction in NLP – A survey.



J. Heinecke and A. Shimorina.

Multilingual abstract meaning representation for celtic languages.

In Proceedings of the 4th Celtic Language Technology Workshop within LREC2022, pages 1–6, 2022.



D. Jurafsky and J. H. Martin.

Speech and language processing (3rd (draft) ed.), 2019.