

LING/C SC 581:

Advanced Computational Linguistics

Lecture 14

Prof. Sandiway Fong

Today's Topics

- Homework 7 Review
 - Human judgements
 - Humans vs. ChatGPT
- Install Stanza on your laptop (Homework 8: Part 1)
- *Crash Blossoms*

Homework 7 Review

- Contrast 1 vs 2 vs 3:
 1. John knew an untrue story about *him*.
 2. John listened to an untrue story about *him*.
 3. John overheard an untrue story about *him*.
- Humans vs. ChatGPT
- Asking ChatGPT the *right* question is key:
 - Who is *him*?
 - Who does *him* refer to?
 - Who is the story about?
 - *Responses may vary. Can we conclude anything?*

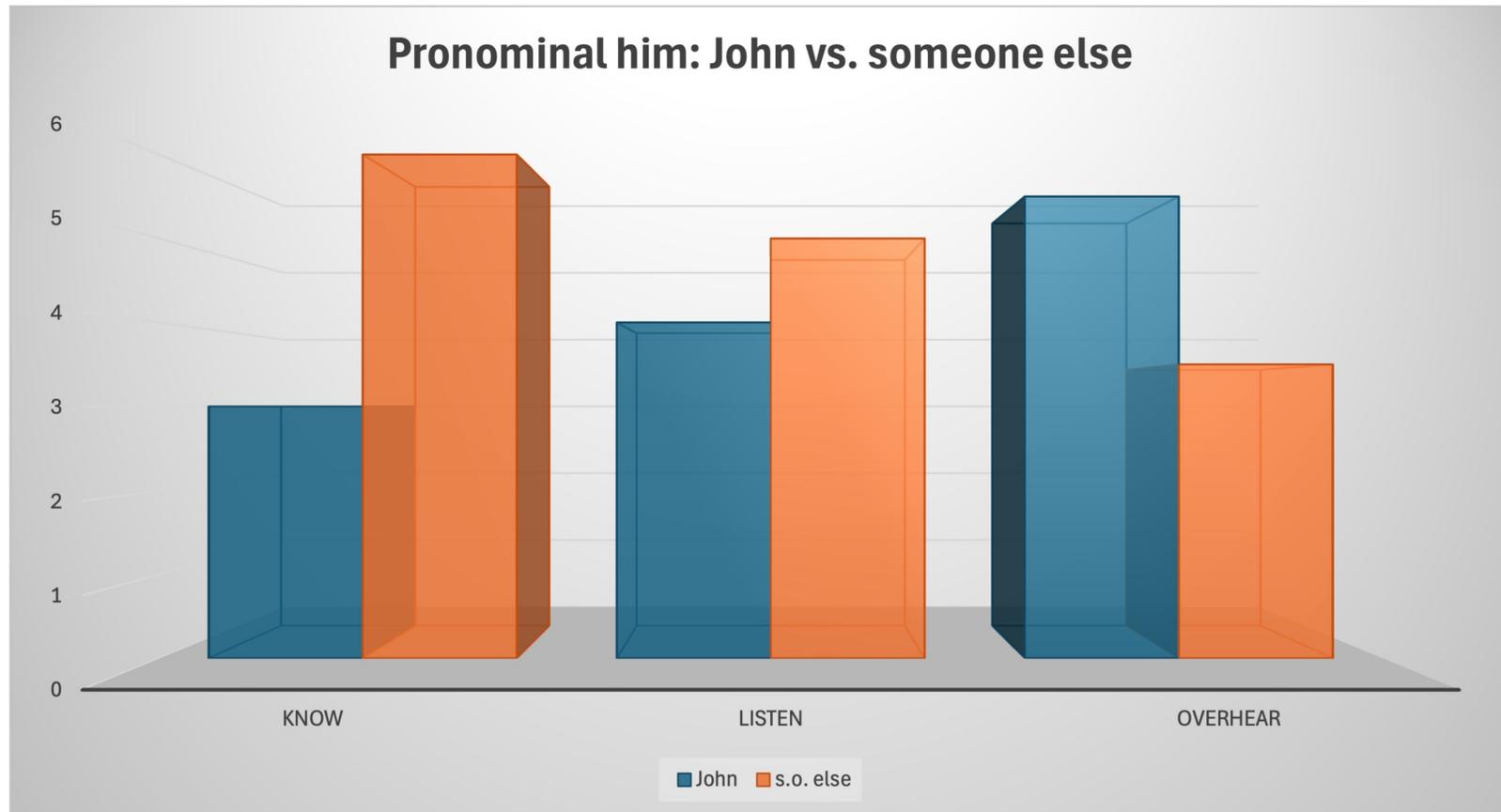
Reflexives and Pronouns

- What is the "algorithm" the human brain uses?
- Examples from Reuland (2010; 93)
 - Max knows which pictures of **himself** Lucie likes
 - Max knows which pictures of **herself** Lucie likes
- *What do you think of these?*
 - Max knows which pictures of **him/ him** Lucie likes
 - Max knows which pictures of **??her/ her** Lucie likes
- *Or these?*
 - Max knows ***him/ him**
 - Max knows **himself**

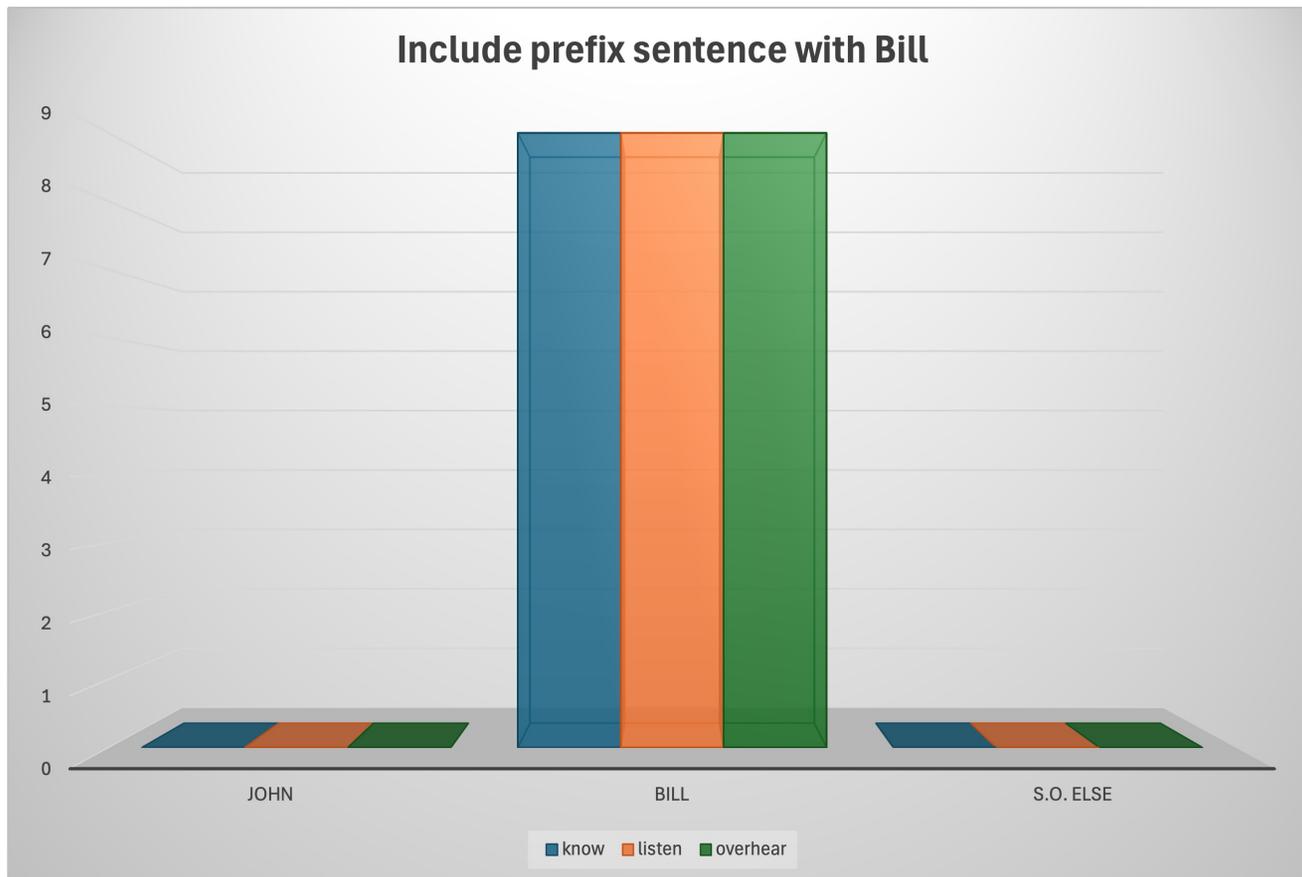
Minimal Search (MS)
upwards

him
her
= someone else

Homework 7 Review



Homework 7 Review



Homework 7 Review: summary

Query	ChatGPT 5.3	ChatGPT 3.5
know + <i>story about</i>	John	John
<i>Bill + ...</i>	Bill	Bill
know + <i>him</i>	John	someone else
<i>Bill + ...</i>	Bill	
listen + <i>story about</i>	John	
<i>Bill + ...</i>	Bill	
listen + <i>him</i>	John	
<i>Bill + ...</i>	Bill	
overhear + <i>story about</i>	John	John
<i>Bill + ...</i>	Bill	Bill
overhear + <i>him</i>	John	
<i>Bill + ...</i>	Bill	

Homework 7 Review: GPT-5.3 vs 3.5

GPT-3.5 vs GPT-5.3 (conceptual comparison)		
Feature	GPT-3.5	GPT-5.3 (modern GPT-5 generation)
Release era	2022	~2024
Reasoning ability	Moderate	Much higher
Context memory	Small-medium	Much higher
Modalities	Text only	Multimodal (text, images, etc.)
Accuracy	Higher hallucination rate	Lower hallucination rate
Coding ability	Basic-integrated	Advanced
Tool usage	Limited	Advanced
Conversation coherence	Sometimes	High

✅ **Short takeaway**
 GPT-3.5 was mainly a **language generator**, while GPT-5-level models are **reasoning systems that can plan, analyze, and use tools**.

GPT-3.5 is believed to be built around **~175 billion parameters**, while later models (starting with GPT-4 and beyond) scale toward **hundreds of billions to trillions of parameters**, increasing capability and knowledge representation. IPL +1

More parameters + better training = better language understanding.

Homework 7 Review: *know + story about*

ChatGPT ▾

John knew an untrue story about him. Who is the story about?

The story is about **John**.

ChatGPT 3.5 ▾



You

John knew an untrue story about him. Who is the story about?



ChatGPT

The statement "John knew an untrue story about him" suggests that John is aware of a false story circulating about himself. Therefore, the story is about John himself.

However, without further context, it's unclear what the specific untrue story might be or what it entails.



Homework 7 Review: *know + him*

ChatGPT ▾

John knew an untrue story about him. Who does him refer to?

In the sentence "John knew an untrue story about him," the pronoun "him" refers to John himself.

ChatGPT 3.5 ▾



You

John knew an untrue story about him. Who does him refer to?



ChatGPT

In the phrase "John knew an untrue story about him," the word "him" likely refers to someone other than John. It suggests that there is a story circulating about a person other than John, and John is aware of this false narrative. Without further context, it's difficult to determine exactly who "him" refers to. It could be a friend, acquaintance, or someone else entirely.



Homework 7 Review: *Bill + know + story about*

ChatGPT ▾

Bill is an honest man. John knew an untrue story about him. Who is the story about?

The story is about **Bill**.

ChatGPT 3.5 ▾



You

Bill is an honest man. John knew an untrue story about him. Who is the story about?



ChatGPT

The story is about Bill. John knew an untrue story about Bill.



Stanza

- <https://stanfordnlp.github.io/stanza/>

 **Stanza** 🔍 Search Stanza

Overview

- Usage
- Neural Pipeline
- Models
- Biomedical Models
- Training
- Stanford CoreNLP Client
- Resources

Stanza – A Python NLP Package for Many Human Languages

`pypi v1.11.1` `conda v1.5.0` `python 3.9 | 3.10 | 3.11 | 3.12 | 3.13`

Stanza is a collection of accurate and efficient tools for the linguistic analysis of many human languages. Starting from raw text, Stanza divides it into sentences and words, and then can recognize parts of speech and entities, do syntactic analysis, and more. Stanza brings state-of-the-art NLP models to languages of your choosing.

Stanza

- <https://stanza.stanford.edu>
- online and off-line (Python package support)

 Stanza 1.11.0 (updated October 2025)

— Text to annotate —

e.g., The quick brown fox jumped over the lazy dog.

— Annotations —

parts-of-speech × named entities × lemmas × dependency parse × constituency parse ×

— Language —

English ▾

Submit

Stanza (*last year*)

```
$ pip3 install stanza
```

```
Collecting stanza
```

```
  Downloading stanza-1.5.0-py3-none-any.whl (802 kB)
```

```
    |████████████████████████████████████████| 802 kB 2.6 MB/s
```

```
...
```

```
Successfully built emoji
```

```
Installing collected packages: torch, protobuf, emoji, stanza
```

```
Successfully installed emoji-2.2.0 protobuf-4.22.1 stanza-1.5.0  
torch-2.0.0
```

Stanza: download model

- <https://stanfordnlp.github.io/stanza/#getting-started>

- Run:

```
$ python3
```

```
Python 3.13.1 (v3.13.1:06714517797, Dec 3 2024, 14:00:22) [Clang 15.0.0 (clang-1500.3.9.4)] on darwin  
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import stanza
```

```
>>> stanza.download('en')
```

```
Downloading https://raw.githubusercontent.com/stanfordnlp/stanza-resources/main/resources_1.11.0.json: 439kB [00:00,  
22.8MB/s]
```

```
2026-03-15 18:20:44 INFO: Downloaded file to /Users/sandiwai/Library/Caches/stanza/1.11.0/resources/resources.json
```

```
2026-03-15 18:20:44 INFO: Downloading default packages for language: en (English) ...
```

```
Downloading https://huggingface.co/stanfordnlp/stanza-en/resolve/v1.11.0/models/
```

```
2026-03-15 18:21:51 INFO: Downloaded file to /Users/sandiwai/Library/Caches/stanza/1.11.0/resources/en/default.zip
```

```
2026-03-15 18:21:52 INFO: Finished downloading models and saved to  
/Users/sandiwai/Library/Caches/stanza/1.11.0/resources
```

```
[['zip', 'default.zip']]
```

```
>>>
```

Stanza: initialize English pipeline

```
>>> nlp = stanza.Pipeline('en')
2026-03-15 18:24:36 INFO: Checking for updates to resources.json in case models have been updated. Note: this behavior can be turned off with download_method=None or download_method=DownloadMethod.REUSE_RESOURCES
2026-03-15 18:24:37 INFO: Downloaded file to /Users/sandaway/Library/Caches/stanza/1.11.0/resources/resources.json
2026-03-15 18:24:38 INFO: Loading these models for language: en (English):
=====
| Processor | Package |
-----
| tokenize | combined |
| mwt | combined |
| pos | combined_charlm |
| lemma | combined_nocharlm |
| constituency | ptb3-revised_charlm |

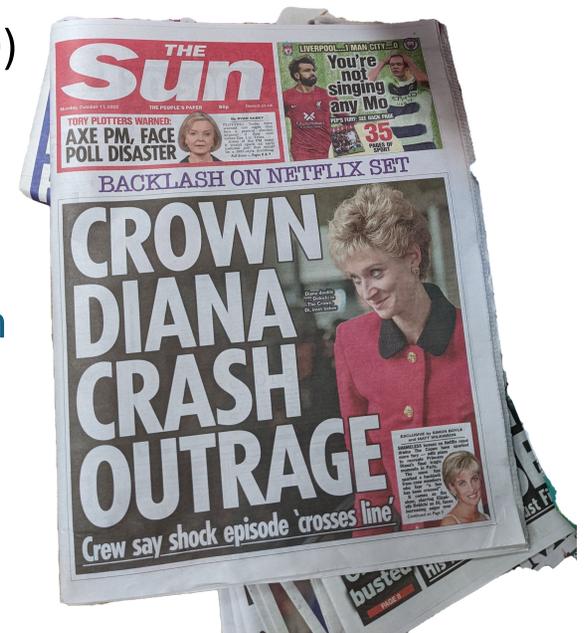
| depparse | combined_charlm |
| sentiment | sstplus_charlm |
| ner | ontonotes-ww-multi_charlm |
=====
2026-03-15 18:24:38 INFO: Using device: cpu
2026-03-15 18:24:38 INFO: Loading: tokenize
2026-03-15 18:24:39 INFO: Loading: mwt
2026-03-15 18:24:39 INFO: Loading: pos
2026-03-15 18:24:40 INFO: Loading: lemma
2026-03-15 18:24:40 INFO: Loading: constituency
2026-03-15 18:24:41 INFO: Loading: depparse
2026-03-15 18:24:41 INFO: Loading: sentiment
2026-03-15 18:24:41 INFO: Loading: ner
2026-03-15 18:24:43 INFO: Done loading processors!
>>>
```

Crash Blossoms

- **Headlines:**

- headlines are compressed (*due to a lack of space*)
- Leaving out words, e.g. copulas etc., may lead to unfortunate structural ambiguities
- <https://www.nytimes.com/2010/01/31/magazine/31FOB-onlanguage-t.html>
- Legendary headlines [...] include “Giant Waves Down Queen Mary’s Funnel,” “MacArthur Flies Back to Front” and “Eighth Army Push Bottles Up Germans.”
- The *Columbia Journalism Review* published two anthologies with the classic titles “Squad Helps Dog Bite Victim” and “Red Tape Holds Up New Bridge.”

a tabloid

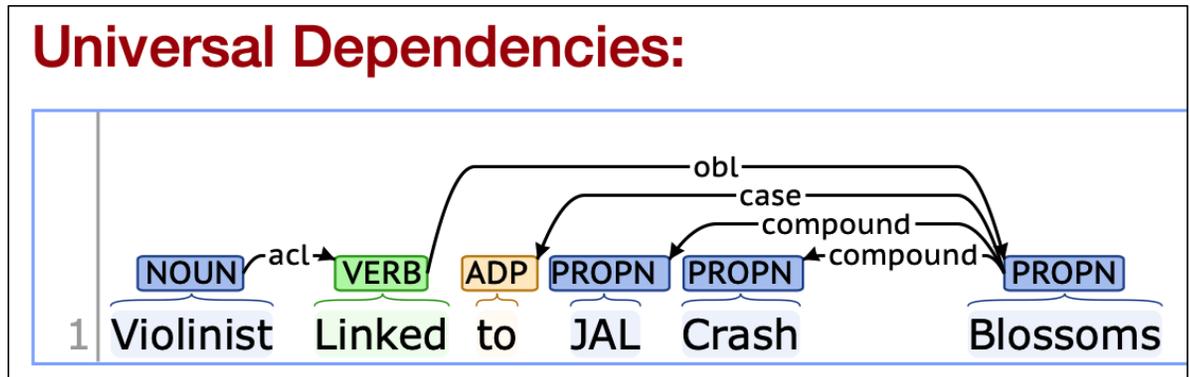


Crash Blossoms

Last August, however, one emerged in the Testy Copy Editors online discussion forum. Mike O’Connell, an American editor based in Sapporo, Japan, spotted the headline “Violinist Linked to JAL Crash Blossoms” and wondered, “What’s a crash blossom?” (The article, from the newspaper Japan Today, described the successful musical career of Diana Yukawa, whose father died in a 1985 Japan Airlines plane crash.)

Violinist Linked to JAL Crash Blossoms

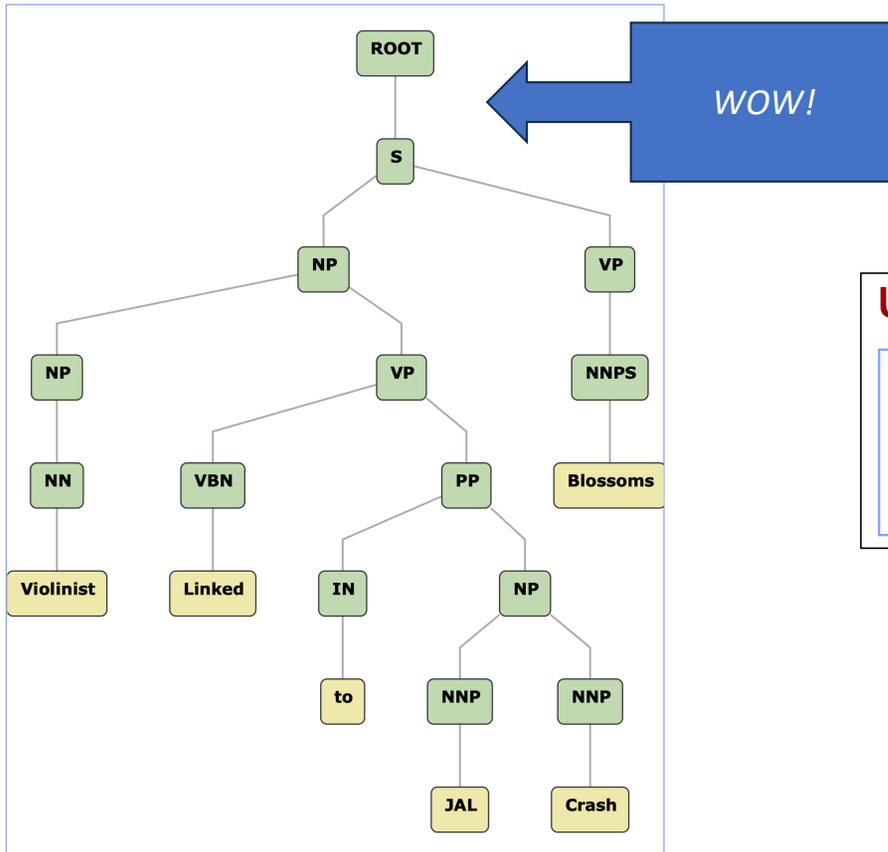
```
>>> doc = nlp("Violinist Linked to JAL Crash Blossoms")
>>> words = doc.sentences[0].words
>>> for word in words:
...     print(f"{word.id}: {word.text} - {word.deprel} -> {word.head}")
...
1: Violinist - root -> 0
2: Linked - acl -> 1
3: to - case -> 6
4: JAL - compound -> 6
5: Crash - compound -> 6
6: Blossoms - obl -> 2
>>>
```



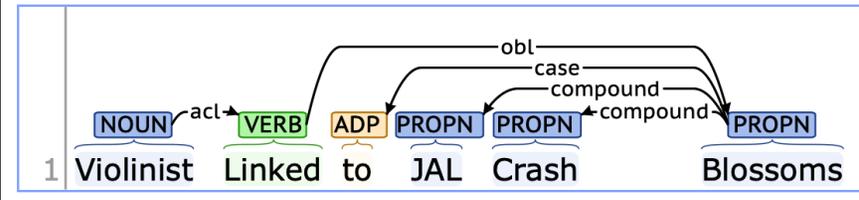
acl: clausal modifier of noun

Violinist Linked to JAL Crash Blossoms

Constituency Parse:



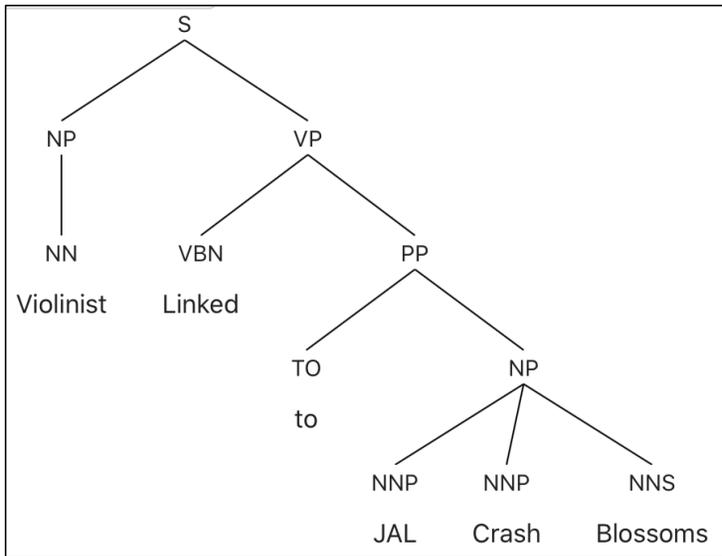
Universal Dependencies:



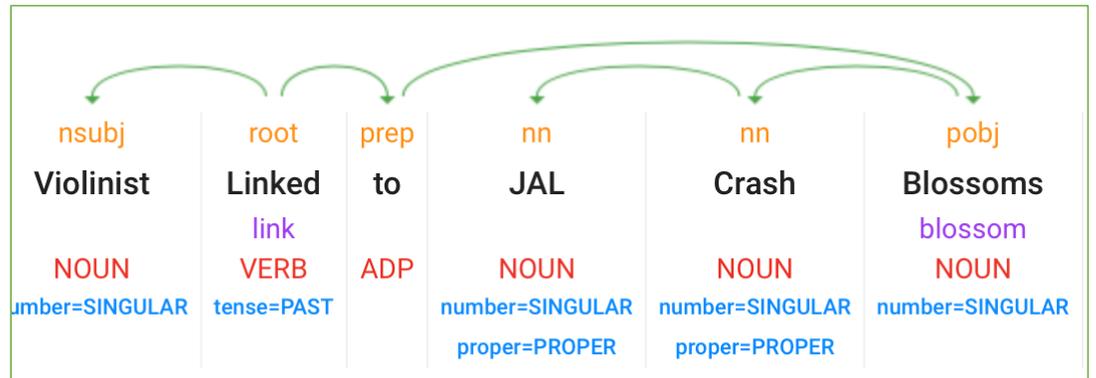
Violinist Linked to JAL Crash Blossoms

```
>>> doc = nlp("Violinist Linked to JAL Crash Blossoms")
>>> print(doc)
[
  [
    {
      "id": 1,
      "text": "Violinist",
      "lemma": "violinist",
      "upos": "NOUN",
      "xpos": "NN",
      "feats":
      "Number=Sing",
      "head": 0,
      "deprel": "root",
      "start_char": 0,
      "end_char": 9,
      "ner": "0",
      "multi_ner": [
        "0"
      ]
    },
    {
      "id": 2,
      "text": "Linked",
      "lemma": "link",
      "upos": "VERB",
      "xpos": "VBN",
      "feats":
      "Tense=Past|VerbForm=Part|Voice=Pass",
      "head": 1,
      "deprel": "acl",
      "start_char": 10,
      "end_char": 16,
      "ner": "0",
      "multi_ner": [
        "0"
      ]
    },
    {
      "id": 3,
      "text": "to",
      "lemma": "to",
      "upos": "ADP",
      "xpos": "IN",
      "head": 6,
      "deprel": "case",
      "start_char": 17,
      "end_char": 19,
      "ner": "0",
      "multi_ner": [
        "0"
      ]
    },
    {
      "id": 4,
      "text": "JAL",
      "lemma": "JAL",
      "upos": "PROPN",
      "xpos": "NNP",
      "feats":
      "Number=Sing",
      "head": 6,
      "deprel": "compound",
      "start_char": 20,
      "end_char": 23,
      "ner": "0",
      "multi_ner": [
        "0"
      ]
    },
    {
      "id": 5,
      "text": "Crash",
      "lemma": "Crash",
      "upos": "PROPN",
      "xpos": "NNP",
      "feats":
      "Number=Sing",
      "head": 6,
      "deprel": "compound",
      "start_char": 24,
      "end_char": 29,
      "ner": "0",
      "multi_ner": [
        "0"
      ]
    },
    {
      "id": 6,
      "text": "Blossoms",
      "lemma": "Blossom",
      "upos": "PROPN",
      "xpos": "NNPS",
      "feats":
      "Number=Plur",
      "head": 2,
      "deprel": "obl",
      "start_char": 30,
      "end_char": 38,
      "ner": "0",
      "multi_ner": [
        "0"
      ],
      "misc":
      "SpaceAfter=No"
    }
  ]
]
```

Violinist Linked to JAL Crash Blossoms



<https://parser.kitaev.io>



pobj : nmod

The pobj relation in SD has been renamed to [nmod](#) in UD.

Note that prepositional phrases are treated differently in UD and SD. See [Prepositional Phrases](#) for more details on the new treatment.

Stanza: Constituency Parse

```
>>> nlp = stanza.Pipeline(lang='en', processors='tokenize,pos,constituency')
2026-03-15 18:59:41 INFO: Checking for updates to resources.json in case models have been updated. Note: this behavior can be turned off with download_method=None or download_method=DownloadMethod.REUSE_RESOURCES
2026-03-15 18:59:41 INFO: Downloaded file to /Users/sandiwai/Library/Caches/stanza/1.11.0/resources/resources.json
2026-03-15 18:59:41 WARNING: Language en package default expects mwt, which has been added
2026-03-15 18:59:42 INFO: Loading these models for language: en (English):
=====
| Processor | Package |
-----
| tokenize | combined |

| mwt | combined |
| pos | combined_charlm |
| constituency | ptb3-revised_charlm |
=====
2026-03-15 18:59:42 INFO: Using device: cpu
2026-03-15 18:59:42 INFO: Loading: tokenize
2026-03-15 18:59:42 INFO: Loading: mwt
2026-03-15 18:59:42 INFO: Loading: pos
2026-03-15 18:59:43 INFO: Loading: constituency
2026-03-15 18:59:44 INFO: Done loading processors!
```

Stanza: Constituency Parse

Description

Constituency parsing is added to the stanza pipeline by using **a shift-reduce parser**.

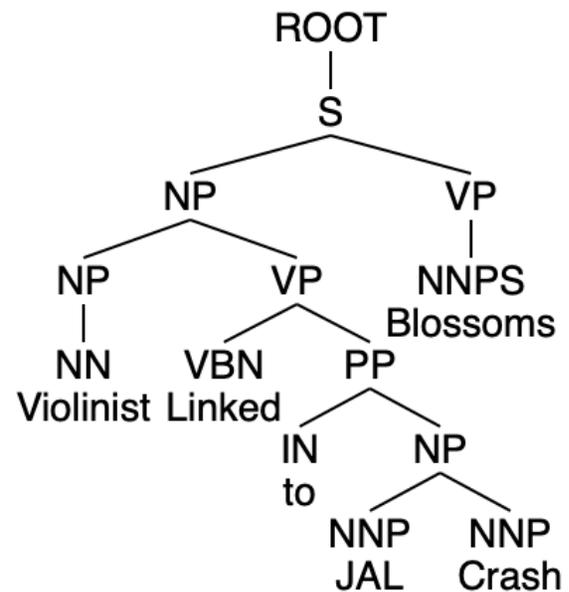
Name	Annotator class name	Requirement	Generated Annotation	Description
constituency	ConstituencyProcessor	tokenize, mwt, pos	constituency	Adds the <code>constituency</code> annotation to each <code>Sentence</code> in the <code>Document</code>

The `ConstituencyProcessor` adds a constituency / phrase structure **parse tree** to each `Sentence`.

Bracket types are dependent on the treebank; for example, the PTB model using the PTB bracket types. Custom models could support any set of labels as long as you have training data.

Stanza: Constituency Parse

```
>>> doc = nlp("Violinist Linked to  
JAL Crash Blossoms")  
>>> for sentence in doc.sentences:  
...     print(sentence.constituency)  
...  
(ROOT (S (NP (NP (NN Violinist)) (VP  
  (VBN Linked) (PP (IN to) (NP (NNP  
  JAL) (NNP Crash))))) (VP (NNPS  
  Blossoms))))  
>>>
```



Stanza: Constituency Parse

- Labels / children

```
>>> tree = doc.sentences[0].constituency
```

```
>>> tree.label
```

```
'ROOT'
```

```
>>> tree.children
```

```
((S (NP (NP (NN Violinist)) (VP (VBN Linked) (PP (IN to) (NP (NNP JAL)  
    (NNP Crash)))))) (VP (NNPS Blossoms))),)
```

```
>>> tree.children[0]
```

```
(S (NP (NP (NN Violinist)) (VP (VBN Linked) (PP (IN to) (NP (NNP JAL) (NNP  
    Crash)))))) (VP (NNPS Blossoms)))
```

```
>>> tree.children[0].label
```

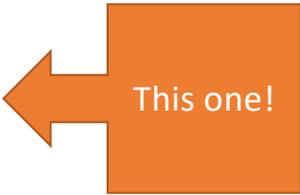
```
'S'
```

```
>>> len(tree.children[0].children)
```

```
2
```

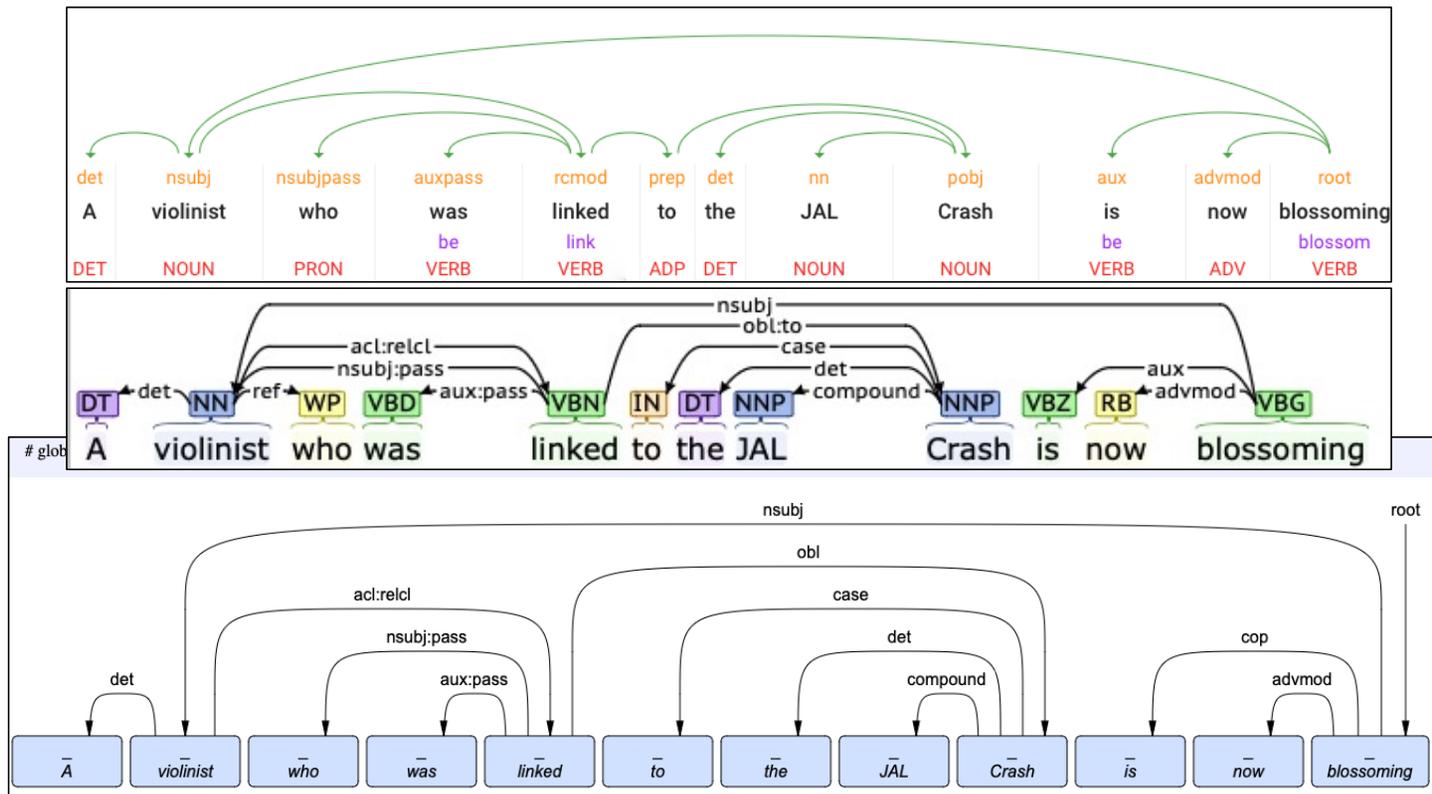
Headlines

- *Violinist Linked to JAL Crash Blossoms*
- *Violinist **Who Was** Linked to JAL Crash Blossoms*
- ***A** Violinist **Who Was** Linked to JAL Crash Blossoms*
- ***A** Violinist **Who Was** Linked to **the** JAL Crash Blossoms*
- ***A** Violinist **Who Was** Linked to **the** JAL Crash **blossoms***
- ***A** **violinist** **who was** **linked** to **the** JAL Crash **blossoms***
- ***A** **violinist** **who was** **linked** to **the** JAL Crash **is now** **blossoming***

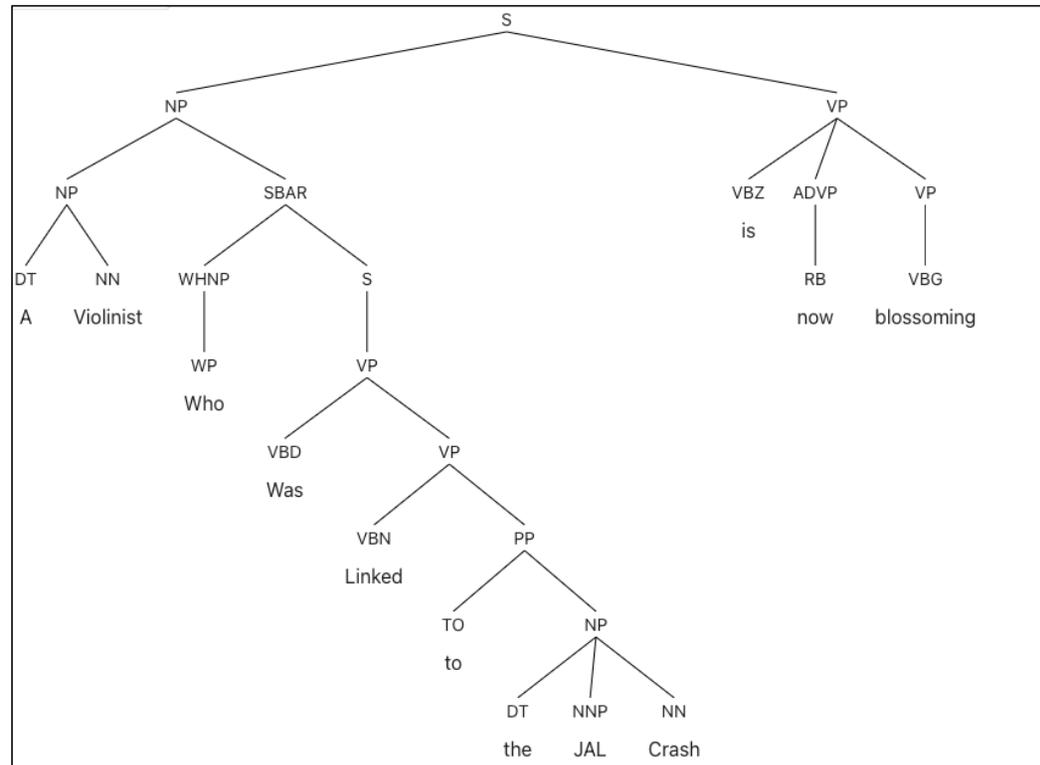


This one!

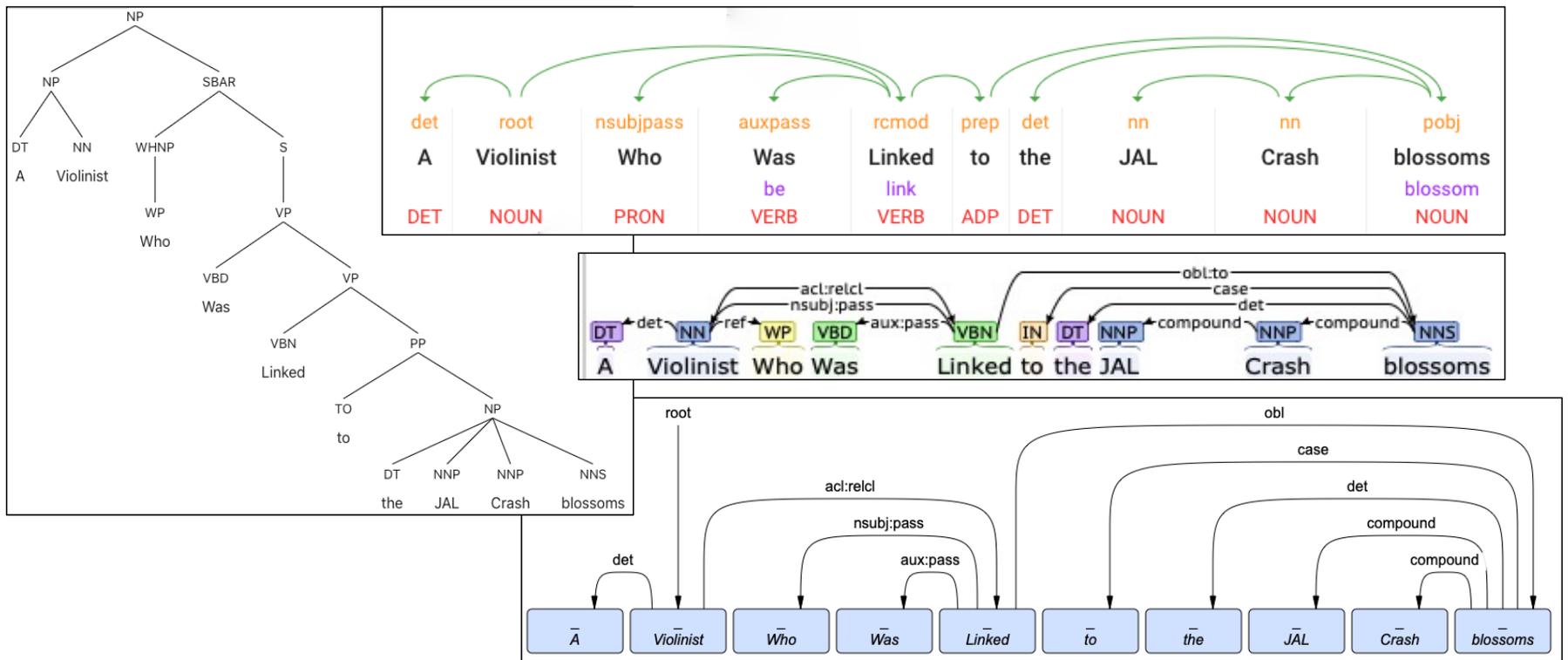
A violinist who was linked to the JAL Crash is now blossoming



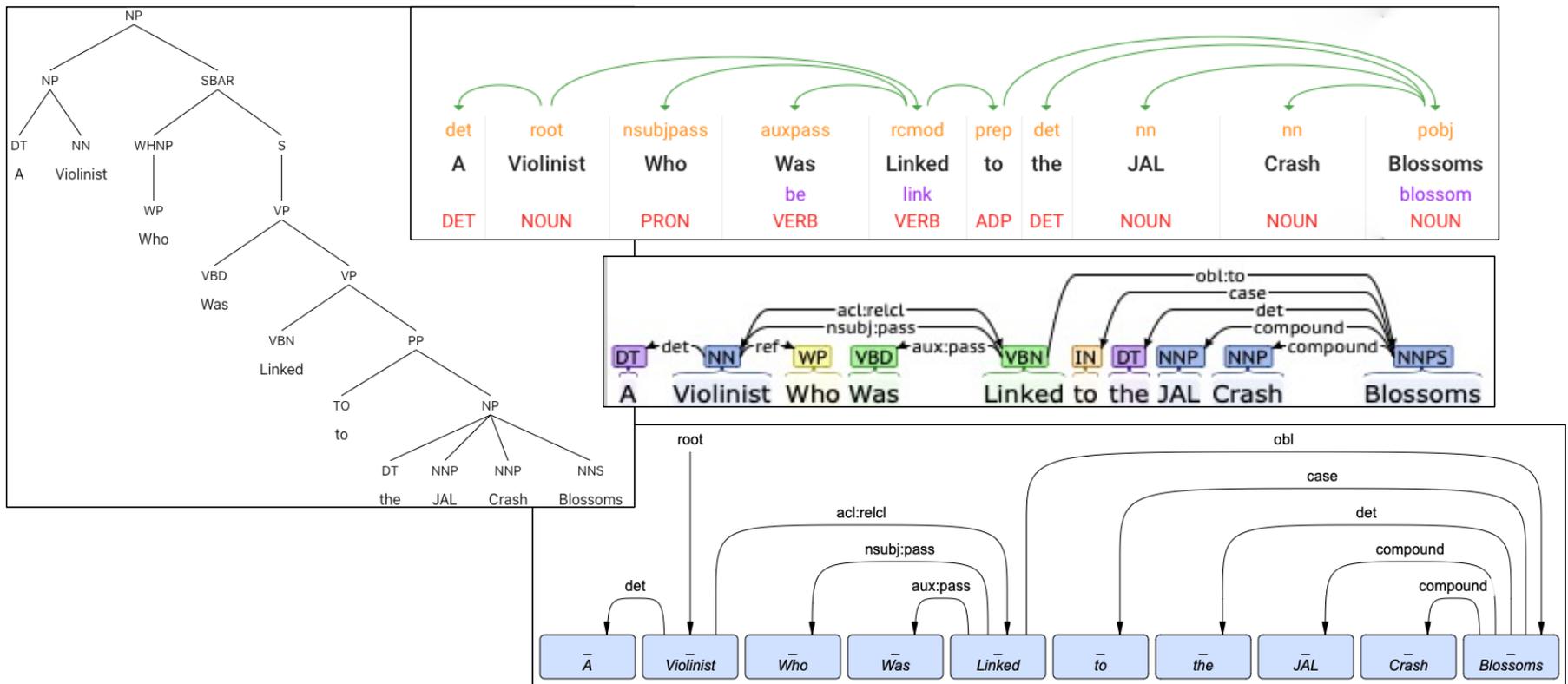
A violinist who was linked to the JAL Crash is now blossoming



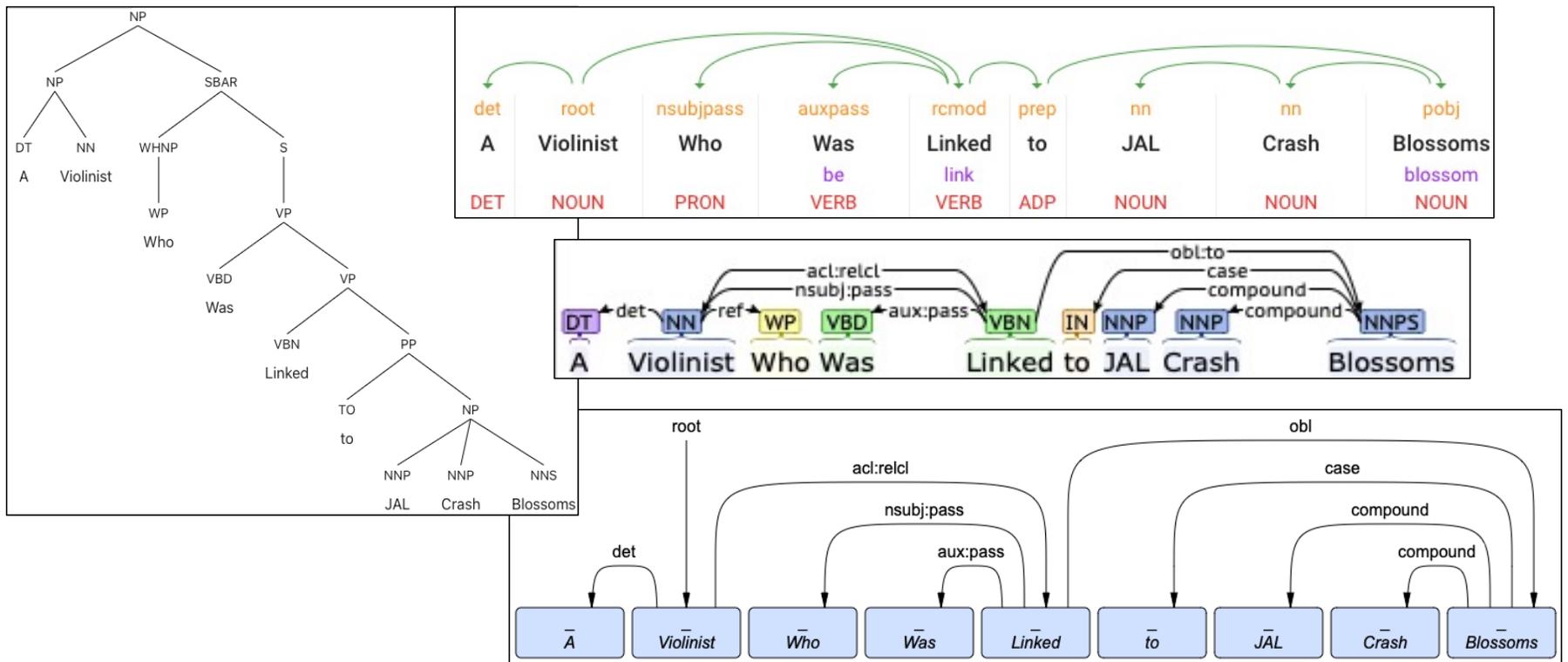
A Violinist Who Was Linked to the JAL Crash blossoms



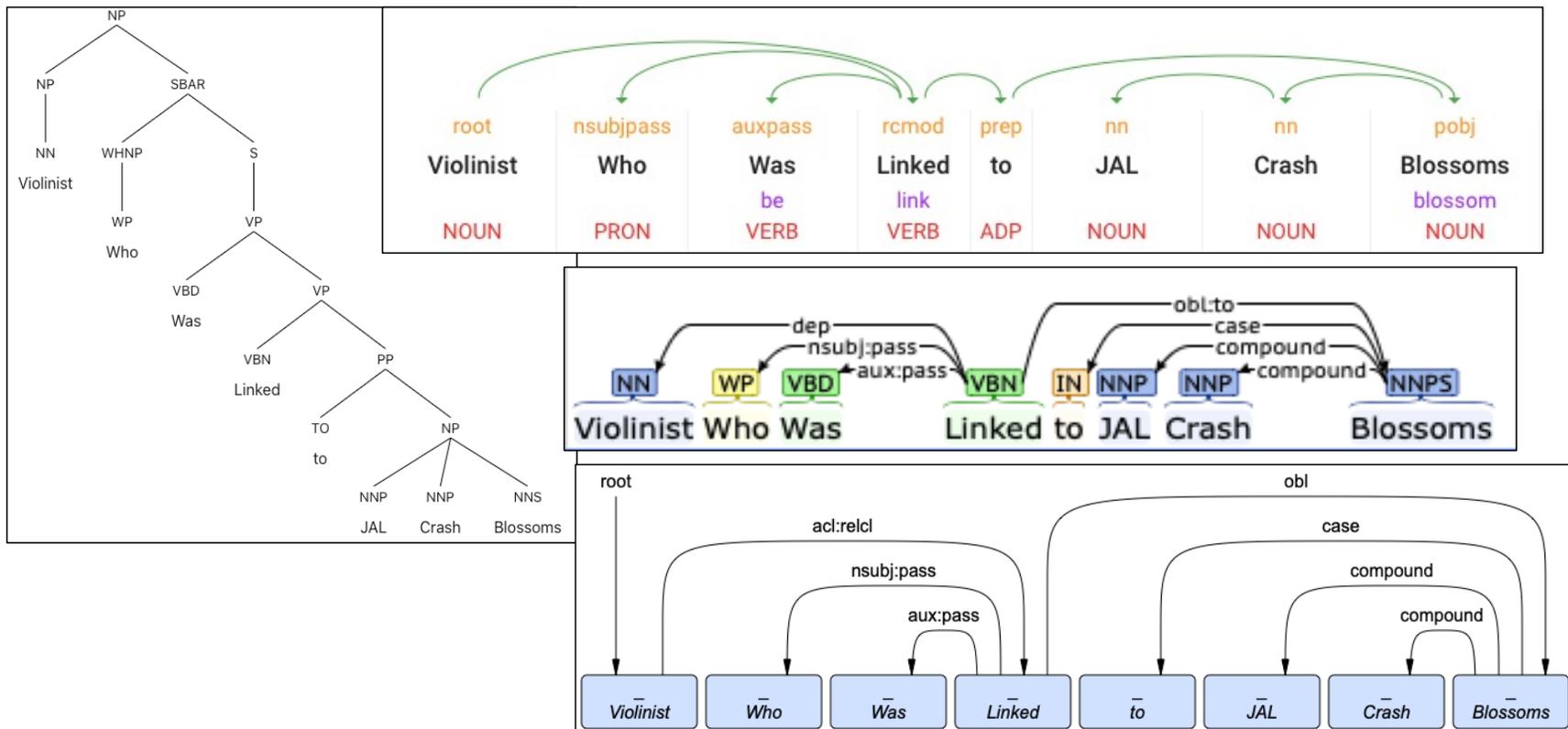
A Violinist Who Was Linked to the JAL Crash Blossoms



A Violinist Who Was Linked to JAL Crash Blossoms



Violinist *Who Was* Linked to JAL Crash Blossoms



Are these Crash Blossoms?

Rich Neville
@RichNeville

Follow

I don't understand how a snake even begins to organise a trip like that.



Entertainment

Lindsay Lohan bitten by snake on holiday in Thailand

< **The Independent's post** ... 🔍

 **The Independent** ✓

4 h · 🌐

Complete list ↓



independent.co.uk

All the easyJet flights cancelled today from UK airports

 **(((Sigourney Beaver)))**
@sigourneybeaver

What a terrible gift.



John Cena surprises 7-year-old boy with cancer on his birthday
winnipegssun.com

- Perhaps not. Funny, yes, but are they the most likely parse/reading?