What Syntactic Information is Necessary for Natural Language Inference?

A comparison of UD and CCG for logic-based NLI

Implementing Natural Language Formal Analysis Workshop 2
2026/06/19

Asa Tomita
Ochanomizu University

The Limitation of UD

The Common Consensus

- Universal Dependencies (UD; Nivre et al., 2016) have standardly enabled cross-linguistic syntactic parsing.
- However, from a formal semantic perspective, **UD inherently lacks sufficient information**

Open Question

While it is widely accepted that UD is "thinner" than grammar formalisms like CCG, **to what exact degree** is this surface-level syntax sufficient or deficient for logical reasoning?

Research Goal

Instead of merely pointing out that "UD lacks information," it is highly meaningful to **quantitatively evaluate** the exact boundaries of its adequacy.

Our Objective

to establish a quantitative benchmark to evaluate syntactic frameworks not by their token-level parsing accuracy, but by their **semantic utility** in high-level language understanding.

Proposed Evaluation Method

Syntax Evaluation from a "Reasoning Perspective"

We propose comparing UD and CCG parsers through the lens of downstream Natural Language Inference (NLI).

The Experimental Pipeline

We measure how the theorem-proving performance of logic-based NLI system shifts when utilizing a UD parser combined with an automatic conversion layer (Tran and Miyao, 2022) to pseudo-CCG derivations.

Ultimate goal

This approach allows us to isolate **how much (and what kind of) syntactic information is truly indispensable** for precise, automated language understanding.

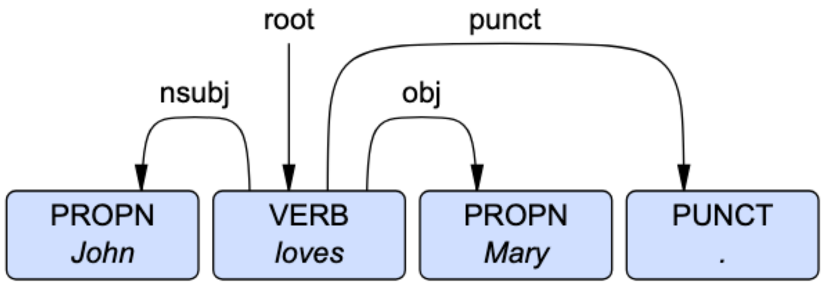
Universal Dependencies (Nivre et al., 2016)

Universal Dependencies (UD)

- a framework for morphosyntactic annotation of human language
- adopts **dependency parsing** as its core syntactic representation
- Scale (as of v2.9): 217 treebanks across 122 languages
- provides a solid basis for the crosslinguistically consistent annotation of typologically diverse languages

Dependency Parsing

- Represents syntax as dependency relations between words
- Designed as a cross-linguistic annotation framework
- Widely available for many languages



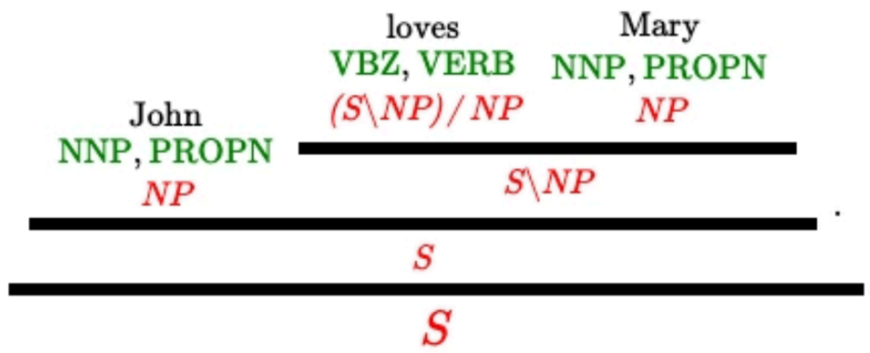
Combinatory Categorial Grammar (CCG; Steedman, 2000)

Highly Lexicalized Phrase Structure Formalism

Each word is assigned a lexical category that specifies its syntactic type and the direction of its arguments.

Combinatory Rules & Semantic Composition

Sentences are parsed through a small set of explicit **combinatory rules** (e.g., Function Application, Composition).



Natural Language Inference (NLI)

Natural Language Inference (NLI) is a task of determining the inferential relation (**entailment**, **contradiction**, or **neutral**) between a premise and a hypothesis.

Entailment : if the premise is true, the hypothesis must also be true.

Premise		Hypothesis
John bought three apples.	$\models$	John bought apples.

Natural Language Inference (NLI)

Natural Language Inference (NLI) is a task of determining the inferential relation (**entailment**, **contradiction**, or **neutral**) between a premise and a hypothesis.

Contradiction : if the premise is true, the hypothesis must be false.

Premise		**Hypothesis**
John bought three apples.	$\nvDash$	John bought no apples.

Natural Language Inference (NLI)

Natural Language Inference (NLI) is a task of determining the inferential relation (**entailment**, **contradiction**, or **neutral**) between a premise and a hypothesis.

Neutral : the truth of the hypothesis cannot be determined from the premise.

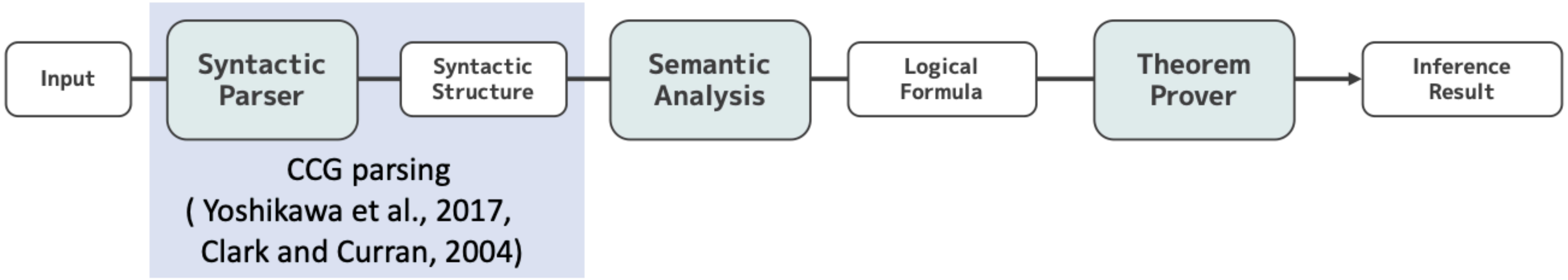
Premise		**Hypothesis**
John bought three apples.	?	John bought a banana.

Logic-based NLI system Pipeline

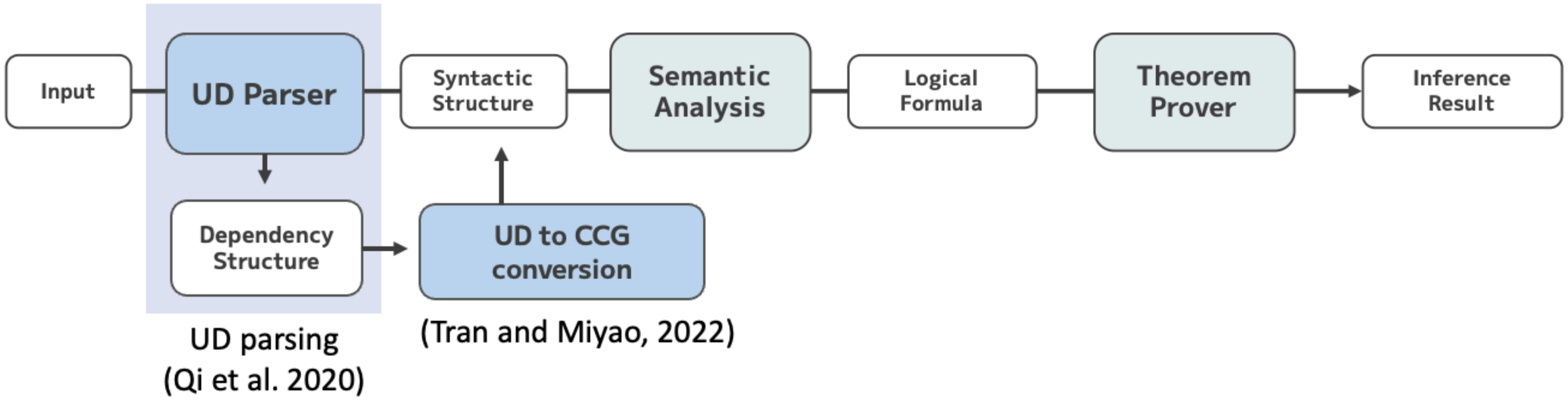
- ccg2lambda (Mineshima et al., 2015), LangPro (Abzianidze, 2017), Boxer (Bos 2015), lightblue (Tomita et al., 2025) are examples of logic-based NLI systems.

CCG/UD Inference Pipeline

- ccg-based pipeline (ccg2lambda)

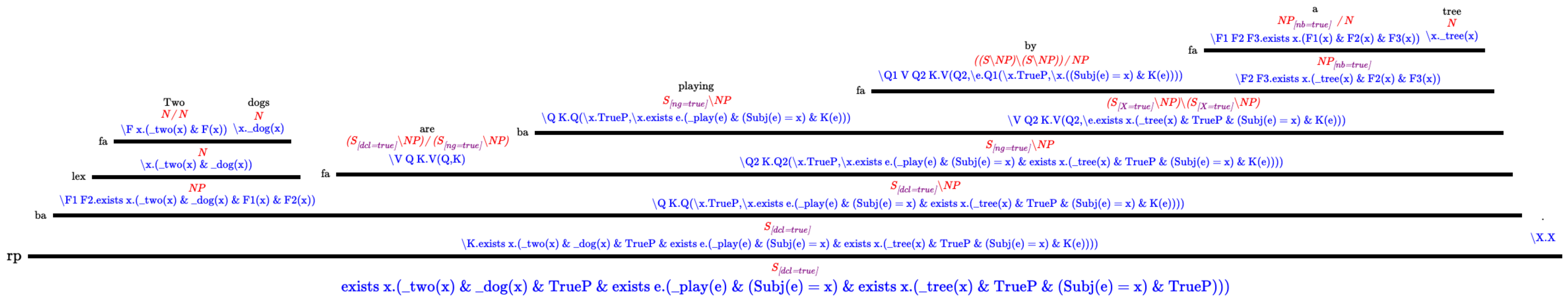


- ud-based pipeline (udparser + ud2ccg-code + ccg2lambda)



ccg2lambda output - CCG derivation

Premise 1: Two dogs are playing by a tree .

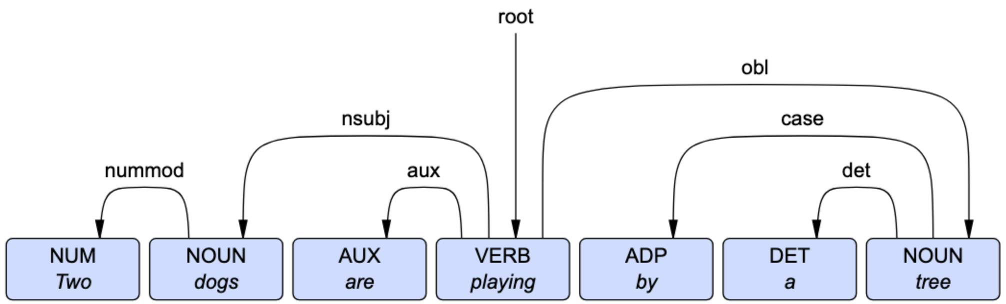


ud-based NLI pipeline - UD parsing

CoNLL-U format

```
# sent_id = 211-1
# entailment_label = ENTAILMENT
# SemEvalSet = TRIAL
# text = Two dogs are playing by a tree
1	Two	two	NUM	CD	NumForm=Word|NumType=Card	2	nummod	_	start_char=0|end_char=3
2	dogs	dog	NOUN	NNS	Number=Plur	4	nsubj	_	start_char=4|end_char=8
3	are	be	AUX	VBP	Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin	4	aux	_	start_char=9|end_char=12
4	playing	play	VERB	VBG	Tense=Pres|VerbForm=Part	0	root	_	start_char=13|end_char=20
5	by	by	ADP	IN	_	7	case	_	start_char=21|end_char=23
6	a	a	DET	DT	Definite=Ind|PronType=Art	7	det	_	start_char=24|end_char=25
7	tree	tree	NOUN	NN	Number=Sing	4	obl	_	start_char=26|end_char=30
```

dependency graph



3. Method

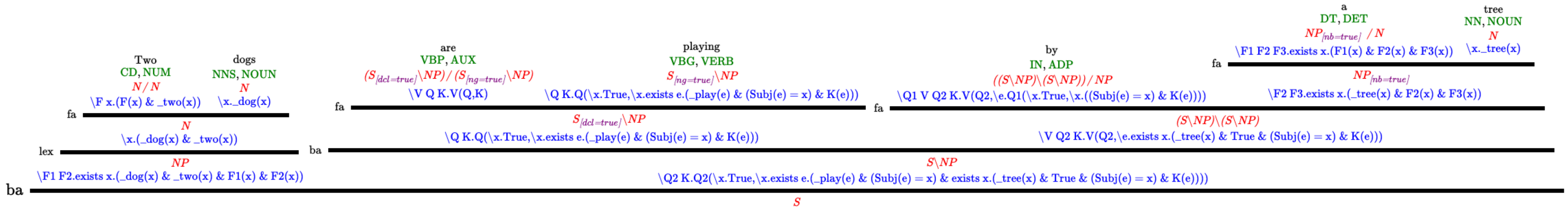
ud-based NLI pipeline - ud2ccg conversion

CCGbank format (`.auto` file)

```
ID=211-1 PARSER=GOLD NUMPARSE=1
(<T S ba 0 2> (<T NP lex 0 1> (<T N fa 0 2> (<L N/N CD NUM Two two (N{103}/N{103}){108}>)
(<L N NNS NOUN dogs dog N{103}>) ) ) (<T S\NP ba 0 2> (<T S\NP fa 0 2>
(<L (S\NP)/(S\NP) VBP AUX are be ((S{102}\NP{103}){102}/(S{102}\NP{103}){102}){114}>)
(<L S\NP VBG VERB playing play (S{102}\NP{103}){102}>) ) (<T (S\NP)\(S\NP) fa 0 2>
(<L ((S\NP)\(S\NP))/NP IN ADP by by
(((S{102}\NP{103}){102}\(S{102}\NP{103}){102}){112}/NP{112}){117}>)
(<T NP fa 0 2> (<L NP/N DT DET a a (NP{119}/N{120}){121}>)
(<L N NN NOUN tree tree N{122}>) ) ) ) )
```

`.auto` file visualizer

Premise 0, tree s0_ccg0: Two dogs are playing by a tree



Recovering category features

CCG feature	UD FEAT	XPOS
`S[dcl]`	VerbForm=Fin	
`S[ng]\NP`	Tense=Pres & VerbForm=Part	
`S[pss]\NP`	Tense=Past & VerbForm=Part	
`NP[nb]/N`		XPOS: DT
`NP[thr]`		EX

Feature recovery helps semantic templates fire correctly.

Experiment setup - SICK dataset

SICK (Sentences Involving Compositional Knowledge; Marelli et al., 2014)

- A benchmark explicitly designed to evaluate compositional distributional semantic models
- 4,500 training, 500 trial, and 4,927 testing pairs (with a stable 29% / 15% / 56% label ratio across all splits).
    - We used 500 trial problems
- Contains less complex linguistic traps, though it demands more robust lexical knowledge.

Overall Results

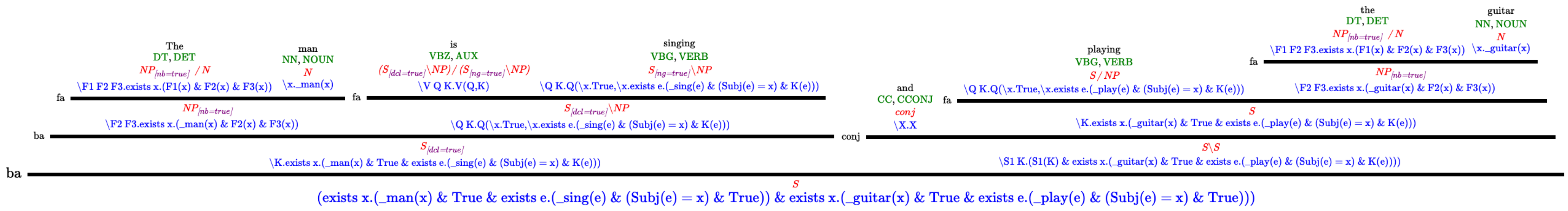
Pipeline Framework	Yes (Gold: 144)	No (Gold: 74)	Unknown (Gold: 282)	Parse/Proof Error
UD-based	79	48	265	20
CCG-based	88	55	269	15

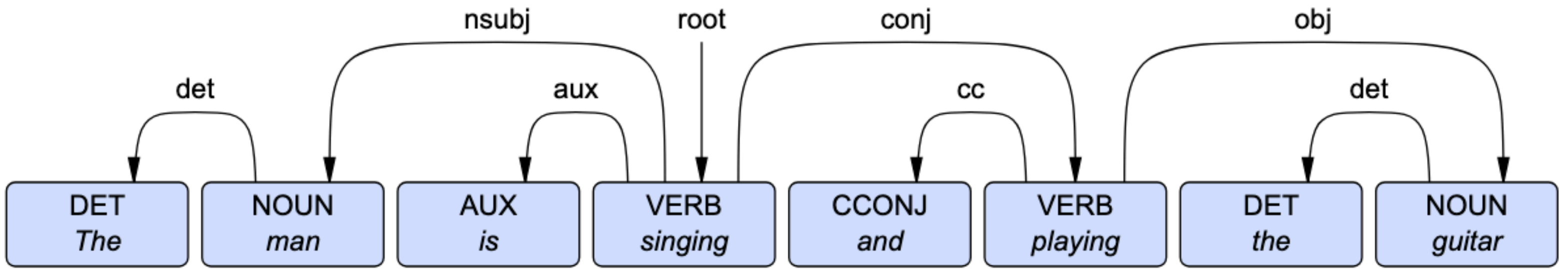
Evaluation Metric	Native CCG	UD-to-CCG	Diff.
Correct Parsing Rate	97.0% (485/500)	96.0% (480/500)	-1.0%
Accuracy	84.4% (422/500)	78.4% (392/500)	-6.0%
Precision	97.9%	97.7%	-0.2%
Recall	65.6%	58.3%	-7.3%
F1 Score	0.7854	0.7299	-0.0555

Error Analysis : Conjoined verbs and verb phrases

SICK ID #3909 [The man is singing] and [playing the guitar].

Premise 0, tree s0_ccg0: The man is singing and playing the guitar





Enhanced Dependencies

Enhanced Universal Dependencies (EUD) enriches the basic UD tree by adding syntactically and semantically useful dependencies.

Basic UD

- Dependency tree
- Each word has (almost always) one head

Enhanced UD

- Dependency graph
- Additional edges can be added
- Makes predicate-argument relations more explicit

Example: coordination

`A woman is singing a song and playing the guitar.`

Problem

The second conjunct may lack an explicit subject.

Semantic parser may fail to introduce the correct event participant.

EUD effect

Enhanced arcs recover the shared subject relation.

Enhanced Dependency: implementation

Enhanced dependencies were obtained from **CoreNLP** (Manning et al.,2014).

We tested two merge strategies:

Strategy	Description
Stanza-based merge	Use Stanza as the base tree and add CoreNLP enhanced arcs
CoreNLP-based merge	Use CoreNLP as the base tree and add Stanza arcs

Enhanced Dependency: implementation

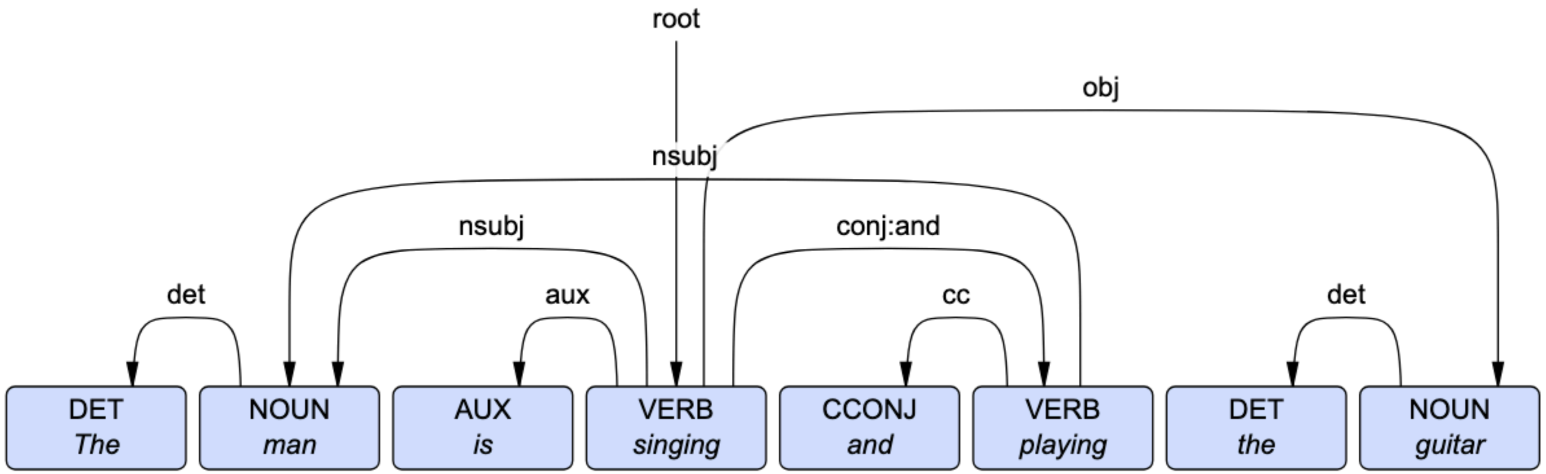
coreNLP

```
# sent_id = 1
# text = The man is singing and playing the guitar
1	The	the	_	DT	_	2	det	2:det	_
2	man	man	_	NN	_	4	nsubj	4:nsubj|6:nsubj	_
3	is	be	_	VBZ	_	4	aux	4:aux	_
4	singing	sing	_	VBG	_	0	ROOT	0:ROOT	_
5	and	and	_	CC	_	6	cc	6:cc	_
6	playing	play	_	VBG	_	4	conj	4:conj:and	_
7	the	the	_	DT	_	8	det	8:det	_
8	guitar	guitar	_	NN	_	4	obj	4:obj	_
```

stanza

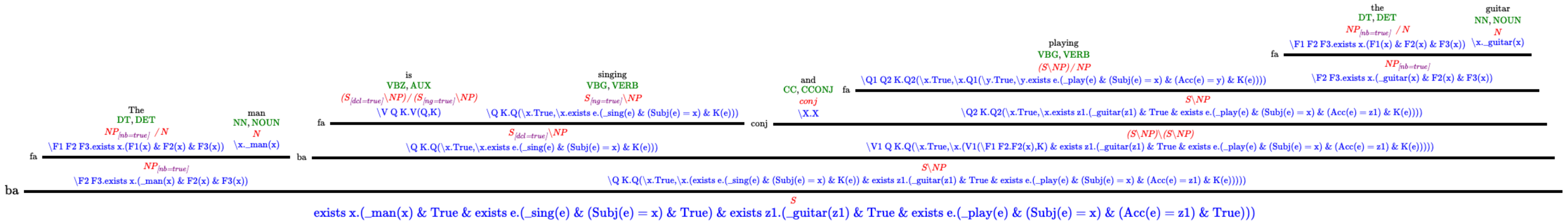
```
# text = The man is singing and playing the guitar
# sent_id = 0
1	The	the	DET	DT	Definite=Def|PronType=Art	2	det	_	start_char=0|end_char=3
2	man	man	NOUN	NN	Number=Sing	4	nsubj	_	start_char=4|end_char=7
3	is	be	AUX	VBZ	Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin	4	aux	_	start_char=8|end_char=10
4	singing	sing	VERB	VBG	Tense=Pres|VerbForm=Part	0	root	_	start_char=11|end_char=18
5	and	and	CCONJ	CC	_	6	cc	_	start_char=19|end_char=22
6	playing	play	VERB	VBG	VerbForm=Ger	4	conj	_	start_char=23|end_char=30
7	the	the	DET	DT	Definite=Def|PronType=Art	8	det	_	start_char=31|end_char=34
8	guitar	guitar	NOUN	NN	Number=Sing	6	obj	_	start_char=35|end_char=41
```

```
# sent_id = 3909-1
# entailment_label = ENTAILMENT
# SemEvalSet = TRIAL
# text = The man is singing and playing the guitar
1	The	the	DET	DT	Definite=Def|PronType=Art	2	det	2:det	start_char=0|end_char=3
2	man	man	NOUN	NN	Number=Sing	4	nsubj	4:nsubj|6:nsubj	start_char=4|end_char=7
3	is	be	AUX	VBZ	Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin	4	aux	4:aux	start_char=8|end_char=10
4	singing	sing	VERB	VBG	Tense=Pres|VerbForm=Part	0	root	0:root	start_char=11|end_char=18
5	and	and	CCONJ	CC	_	6	cc	6:cc	start_char=19|end_char=22
6	playing	play	VERB	VBG	VerbForm=Ger	4	conj	4:conj:and	start_char=23|end_char=30
7	the	the	DET	DT	Definite=Def|PronType=Art	8	det	8:det	start_char=31|end_char=34
8	guitar	guitar	NOUN	NN	Number=Sing	6	obj	4:obj	start_char=35|end_char=41
```



EUD to CCG conversion

Premise 0, tree s0_ccg0: The man is singing and playing the guitar



- The EUD-based conversion generated a valid CCG derivation.

EUD-based NLI experiment

Pipeline Framework	Yes (Gold: 144)	No (Gold: 74)	Unknown (Gold: 282)	Parse/Proof Error
EUD-based	92	53	277	2
UD-based	79	48	265	20
CCG-based	88	55	269	15

Evaluation Metric	CCG-based	UD-based	EUD-based
Correct Parsing Rate	97.0% (485/500)	96.0% (480/500)	99.6% (498/500)
Accuracy	84.4% (422/500)	78.4% (392/500)	84.4% (422/500)
Precision	0.979	0.977	0.979
Recall	0.6560	0.5830	0.6651
F1 Score	0.7854	0.7299	0.7923

Discussion

Basic UD does not always provide sufficient information for logical inference.

However, Enhanced UD recovers much of the missing information and achieves performance comparable to CCG on SICK.

Conclusion

Basic UD contains much of the information needed for logical inference.

However, coordination, relative clauses, scope, and composition often require enriched syntactic information.

Limitation and Future Work

- Larger-scale experiment and error analysis
- More systematic classification of error types
- LangPro evaluation with UD-derived CCG
- Japanese UD→CCG and JSICK
- Other datasets: HANS, FraCaS, Hy-NLI
- Other UD and enhanced dependency parsers