

生成AIにおける Few-Shot 係り受け解析

安岡孝一*

1 きっかけ

2026 年 6 月 1 日から 7 月 13 日にかけて、Qiita Tech Festa 2026 ^{a)}の記事投稿キャンペーンがおこなわれた。いくつかのお題で Qiita 記事を競う、というものだが、この中の「国内 AI モデル推論プラットフォーム「ai& Inference」を使ってみよう」^{b)}が筆者の目を引いた。以下の 9 つの生成 AI を、OpenAI 互換 API で使えるのが魅力的だったので、早速 Qiita 記事^{c)}を書いて「ai& Inference」を使ってみることにした。

deepseek-ai/deepseek-v4-flash

deepseek-ai/deepseek-v4-pro

google/gemma-4-31b-it

moonshotai/kimi-k2.6

moonshotai/kimi-k2.7-code

openai/gpt-oss-120b

qwen/qwen3.6-27b

zai-org/glm-5.1

zai-org/glm-5.2

筆者が班長を務める京都大学人文科学研究所共同研究班「古典中国語コーパスの応用研究」(班員: Christian Wittern・池田巧・李媛・劉冠偉・久保旭・守岡知彦・山崎直樹・二階堂善弘・鈴木慎吾・師茂樹・藤田一乗) とその後継にあたる「古典中国語コーパスの発展的研究」では、古典中国語と周辺言語の係り受け解析に携わっており、その道具立てとして、Universal Dependencies (UD) [1] を各言語に適用している。GPT 系言語モデルによる係り受け解析にも挑戦した [2] のだが、系列ラベリングだとイマイチうまく行かない感じだった。その後に、松田・馬・浅原の手法 [3, 4] を読んで、ならば user-assistant prompt の Few-Shot だと、どの程度まで係り受け解析できるのだろう、という疑問がウズウズと湧いてきたのである。

本稿では「ai& Inference」の助けを借りて、この疑問に挑戦する。

2 Universal Dependencies の概要

UD は、書写言語における品詞・形態素属性・依存構造(係り受け関係)を、言語に関わらず記述する手法である。句構造を考慮せずに係り受け関係を記述することで、言語横断性を高めており、全ての文法構造を単語間のリンクで記述するのが特徴である。

*京都大学人文科学研究所附属人文情報学創新研究センター

^{a)}<https://qiita.com/tech-festa/2026>

^{b)}<https://qiita.com/official-events/750d1f37b7217167b1ad>

^{c)}<https://qiita.com/KoichiYasuoka/items/8b2675a4fca9413a081b>

表 1: CoNLL-U の各フィールド

1. ID: 単語ごとに付与されたインデックスで、文ごとに 1 から始まる整数。縮約語に対しては、単語の範囲を示すのも可。
2. FORM: 語、または、句読記号。
3. LEMMA: 基底形、語幹。
4. UPOS: UD で規定された言語普遍的な品詞タグ (表 2)。
5. XPOS: 言語固有の品詞タグ。
6. FEATS: UD で規定された言語普遍的な形態素属性のリスト。言語固有の拡張も可。
7. HEAD: 当該の単語の係り受け元 ID。係り受け元が無い場合は 0 とする。
8. DEPREL: UD で規定された言語普遍的な係り受けタグ (表 3)。HEAD が 0 の場合は root とする。言語固有の拡張も可。
9. DEPS: 複数の係り受け元を持つ場合、全ての HEAD:DEPREL ペア。
10. MISC: その他のアノテーション。

表 2: UD 品詞タグ (UPOS)

Open class words	Closed class words	Other
ADJ 形容詞 ADV 副詞 INTJ 感嘆詞 NOUN 名詞 PROPN 固有名詞 VERB 動詞	ADP 側置詞 AUX 助動詞 CCONJ 並列接続詞 DET 限定詞 NUM 数詞 PART 接辞 PRON 代名詞 SCONJ 従属接続詞	PUNCT 句読点 SYM 記号 X その他

表 3: UD 係り受けタグ (DEPREL)

	Nominals	Clauses	Modifier Words	Function Words
Core arguments	nsubj 主語 obj 目的語 iobj 間接目的語	csubj 節主語 ccomp 節目的語 xcomp 節補語		
Non-core dependents	obl 斜格補語 vocative 呼称語 expl 形式語 dislocated 外置語	advcl 連用修飾節	advmod 連用修飾語 discourse 談話要素	aux 動詞補助成分 cop 繫辞 mark 標識
Nominal dependents	nmod 体言による連体修飾語 appos 同格 nummod 数量による修飾語	acl 連体修飾節	amod 用言による連体修飾語	det 決定語 clf 類別語 case 格表示
Coordination	MWE	Loose	Special	Other
conj 接続 cc 接続語	fixed 固着 flat 並列 compound 複合	list 細目 parataxis 隣接表現	orphan 親なし goeswith 泣き別れ reparandum 言い損じ	punct 句読点 root 親 dep 未定義

依存構造解析それ自体は、Tesnière の構造的統語論 [5] に源を発し、Мельчук の有向グラフ記述 [6] によって、一応の完成を見た手法である。その最大の特長は、いわゆる動詞中心主義によって言語横断的な記述が可能だという点にあり、Мельчук 依存文法をコンピュータ向けに洗練した UD においても、言語に関わらない記述、という特長が前面に押し出されている。UD における文法構造記述は、句構造を考慮せず、全てを単語間のリンクとして表現する。これにより、言語横断的な文法構造記述を可能としている。

UD 係り受けコーパスの交換用フォーマットとして、CoNLL-U と呼ばれるタブ区切りテキスト (文字コードは UTF-8) が規定されている。CoNLL-U の各行は各単語に対応しており、表 1 に示す 10 個のタブ区切りフィールドで構成される。ID・FORM・LEMMA は、単語そのものに関するフィールドである。UPOS・XPOS・FEATS は、単語の品詞と形態素属性に関するフィールドである。HEAD・DEPREL・DEPS は、単語の係り受けに関するフィールドである。

UD における係り受け関係は、単語間の有向グラフを HEAD と DEPREL で記述する。HEAD は、その単語に入る有向枝のリンク元 ID を示しており、DEPREL は、その有向枝における係り受けタグである。ただし、HEAD が 0 の場合、その枝に入るリンク元は存在しない。リンクの本数は単語の個数に等しく、各リンクのリンク先は、全て互いに異なっている。すなわち、各単語から出るリンクは複数の可能性があるが、各単語に入るリンクは 1 つだけである。なお、リンクはループしない。

UD の係り受けリンクは、Мельчук 依存文法の後裔にあたり、いわゆる動詞中心主義である。動詞をリンク元として、主語や目的語へとリンクする。修飾関係においては、被修飾語から修飾語へとリンクする。ただし、側置詞 (前置詞や後置詞) を体言の修飾語だとみなす点 [7] が、Мельчук とは異なっている。また、コピュラ文においては動詞中心主義を採らず、補語をリンク元として、主語や繫辞へとリンクする。

日本語 UD の例として、「最初の例文」の CoNLL-U と deplacy [8] による可視化を、図 1 に示す。

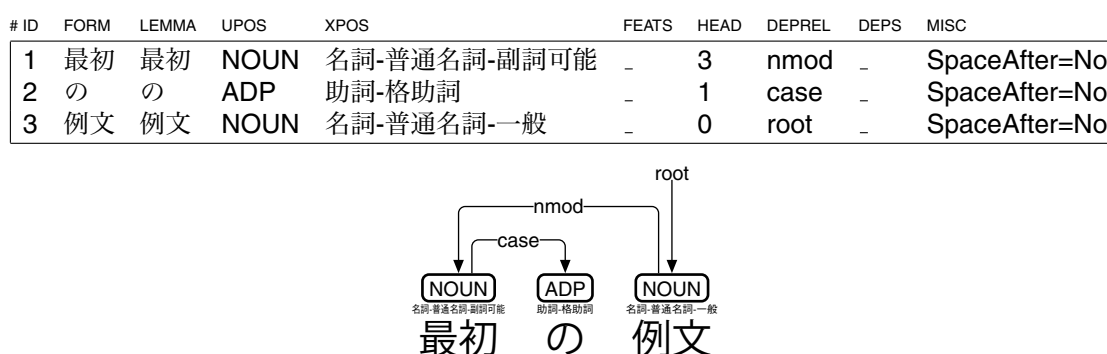


図 1: 「最初の例文」の CoNLL-U と可視化

3 生成 AI における Few-Shot 係り受け解析

CoNLL-U の各行を、FORM・UPOS・DEPREL・リンク元 FORM、の 4 つ組で表し、さらに 1 行にまとめることを考える。たとえば、図 1 の「最初の例文」に対しては

最初_NOUN_nmod_例文|の_AD case_最初|例文_NOUN_root_例文

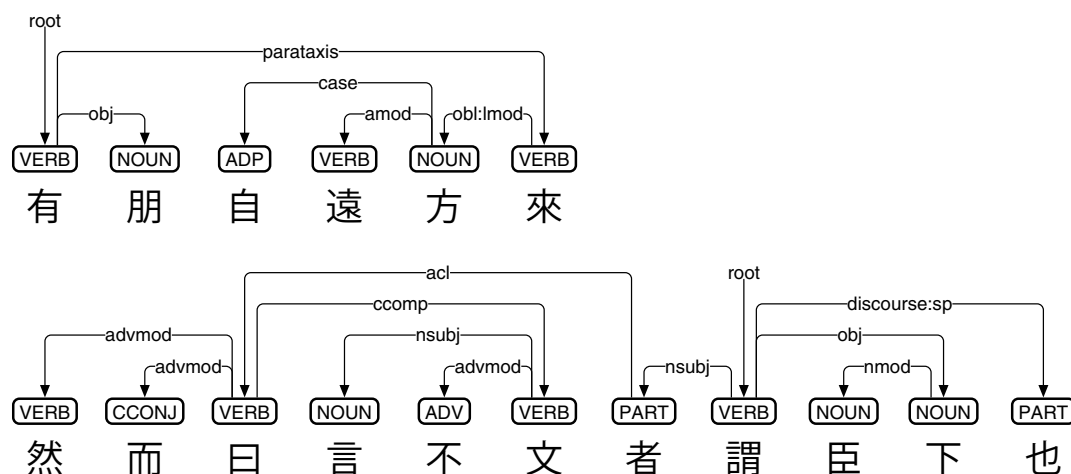
とまとめる。なお、**root** のリンク元は自分自身とする。これを assistant prompt とみなし、user-assistant 形式で例文を 3 つ並べた後、最初の system prompt に「Try dependency-parsing with UPOS and DEPREL.」を、最後の user prompt に「係り受け解析したい文」を付け加えたのが、図 2 のプログラムである。

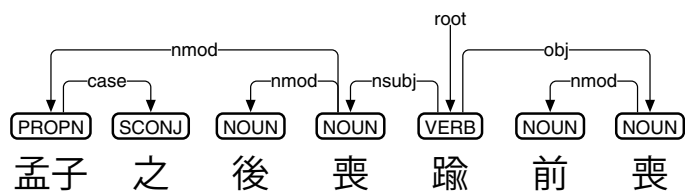
図 2 のプログラムを雛型として、各言語での Few-Shot 係り受け解析に挑戦してみよう。

```
#!/usr/bin/python3
msg=[{"role":"system","content":"Try dependency-parsing with UPOS and DEPREL."},
      {"role":"user","content":"最初の例文"}, {"role":"assistant","content":
      "最初_NOUN_nmod_例文|の_ADP_case_最初|例文_NOUN_root_例文"},
      {"role":"user","content":"次の例文"}, {"role":"assistant","content":
      "次_NOUN_nmod_例文|の_ADP_case_次|例文_NOUN_root_例文"},
      {"role":"user","content":"その次の例文"}, {"role":"assistant","content":
      "その_DET_det_次|次_NOUN_nmod_例文|の_ADP_case_次|例文_NOUN_root_例文"},
      {"role":"user","content":"係り受け解析したい文"}]
url="https://api.aiand.com/v1"
key="sk-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
mdl="moonshotai/kimi-k2.6"
if url==None:
    from transformers import pipeline
    nlp=pipeline("text-generation",mdl,temperature=0.00001,token=key)
    res=nlp(msg)
    print(res[0]["generated_text"][-1]["content"].strip().replace("\n","|"))
else:
    from openai import OpenAI
    cli=OpenAI(base_url=url,api_key=key)
    res=cli.chat.completions.create(model=mdl,temperature=0,messages=msg)
    print(res.choices[0].message.content.strip().replace("\n","|"))
```

図 2: Few-Shot 係り受け解析プログラム (雛型)

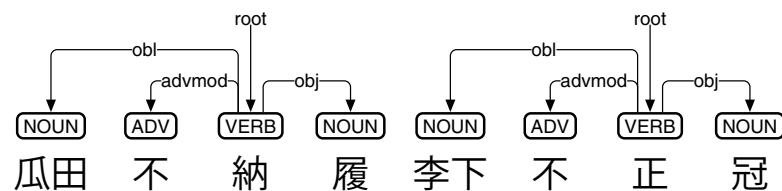
3.1 古典中国語 Few-Shot 係り受け解析



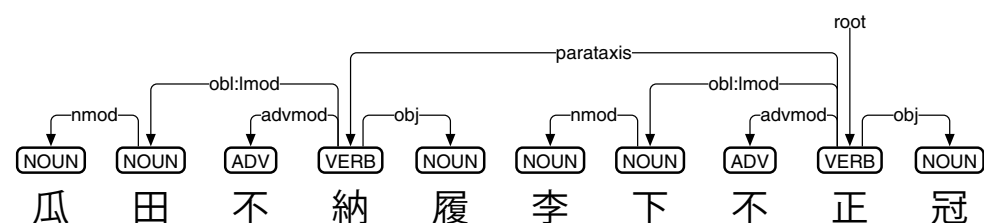


上記3つを例文として、「瓜田不納履李下不正冠」の Three-Shot 係り受け解析に挑戦してみた。

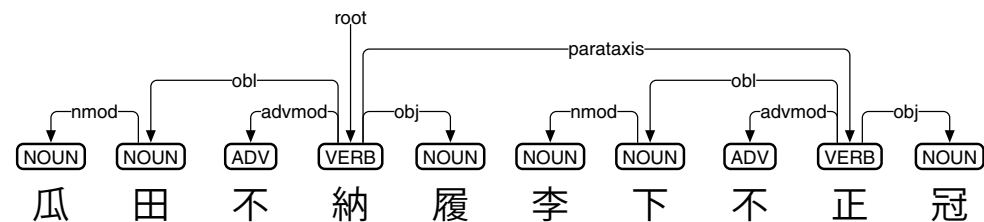
deepseek-ai/deepseek-v4-flash



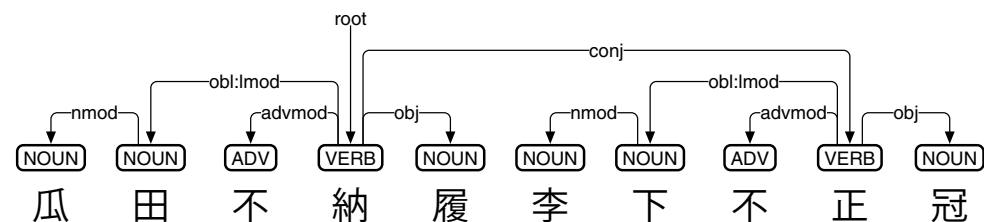
deepseek-ai/deepseek-v4-pro



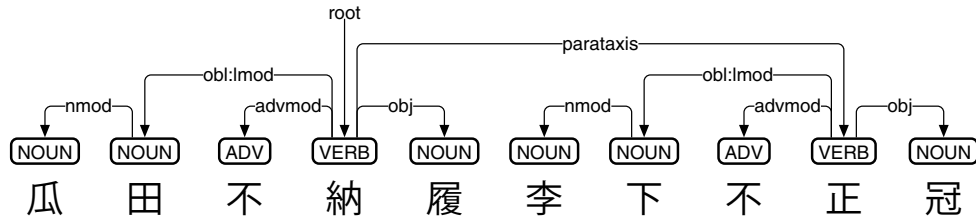
google/gemma-4-31b-it



moonshotai/kimi-k2.6

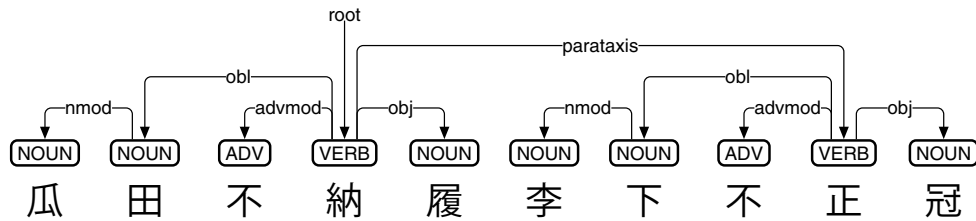


moonshotai/kimi-k2.7-code

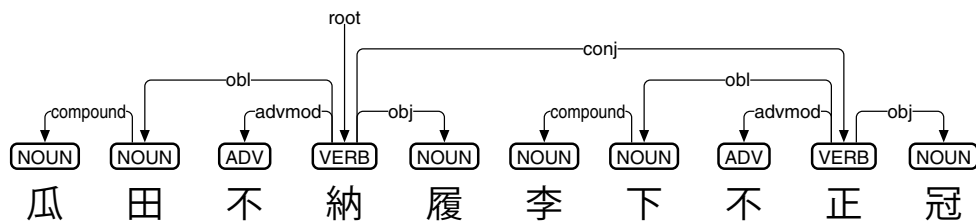


openai/gpt-oss-120b はリンク元が全く出力されておらず、フォーマットが壊れている。

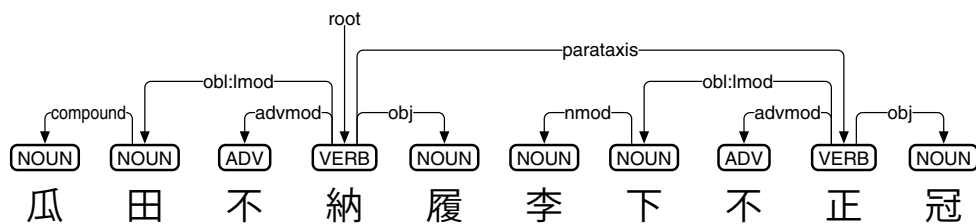
qwen/qwen3.6-27b



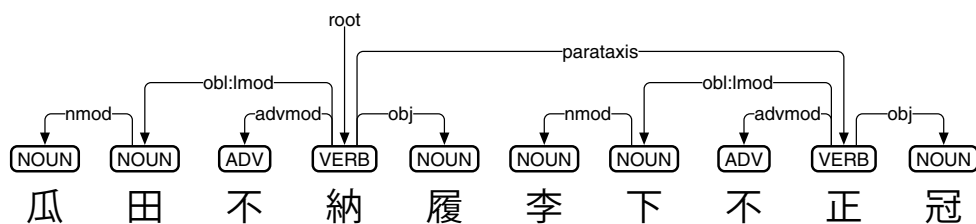
zai-org/glm-5.1



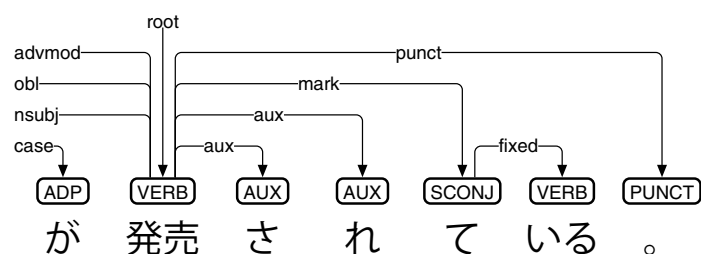
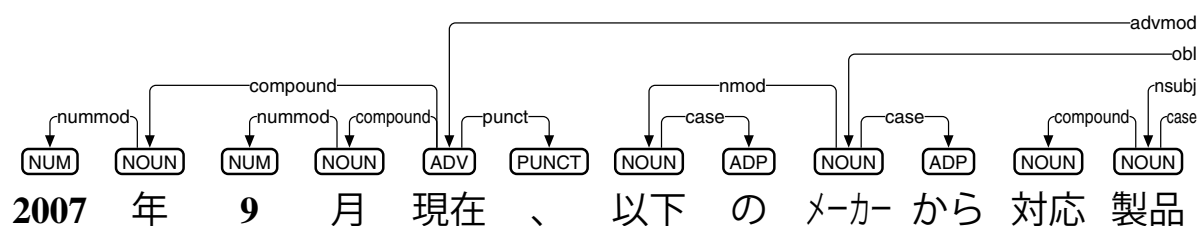
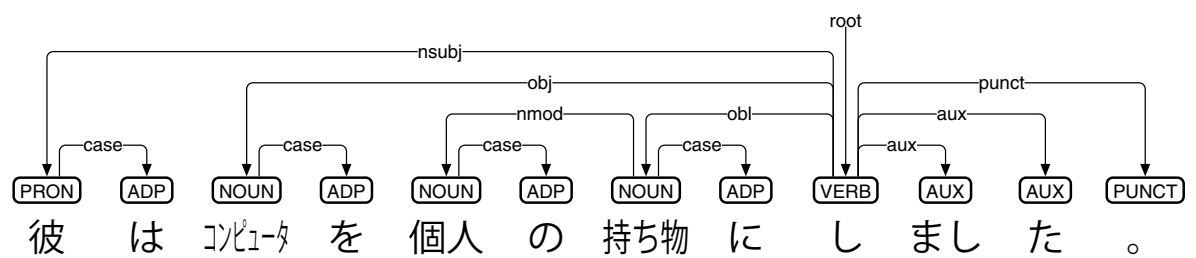
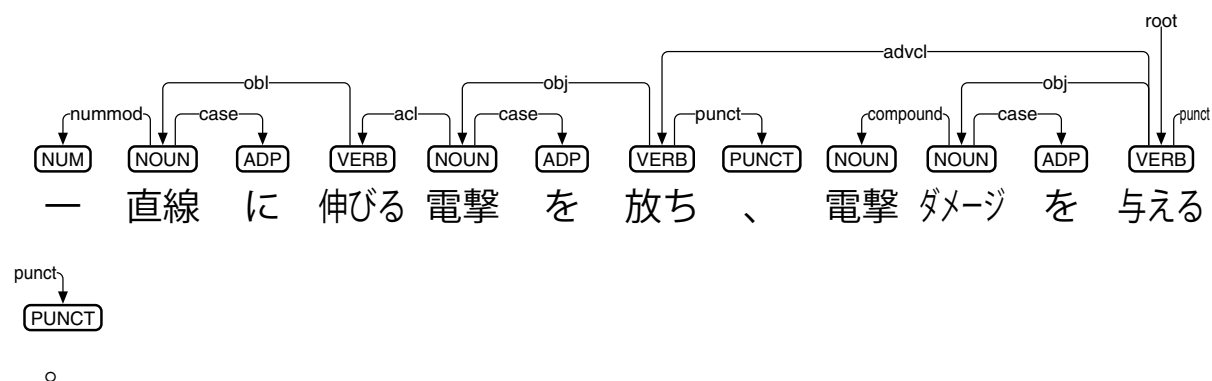
zai-org/glm-5.2



筆者の考える正解

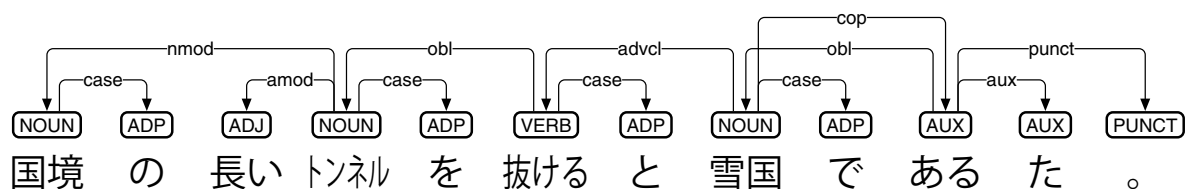


3.2 現代日本語 Few-Shot 係り受け解析



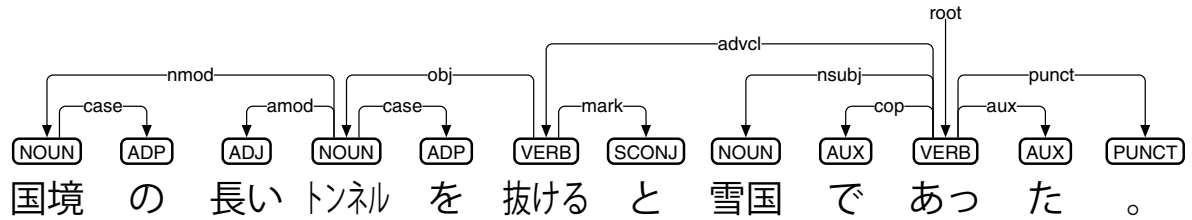
上記3つを例文として、「国境の長いトンネルを抜けると雪国であった。」の Three-Shot 係り受け解析に挑戦してみた。

deepseek-ai/deepseek-v4-flash

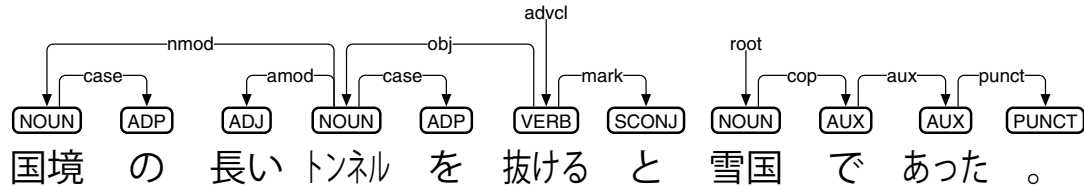


「あっ」が「ある」に化けており、しかもリンクがループしている。

deepseek-ai/deepseek-v4-pro

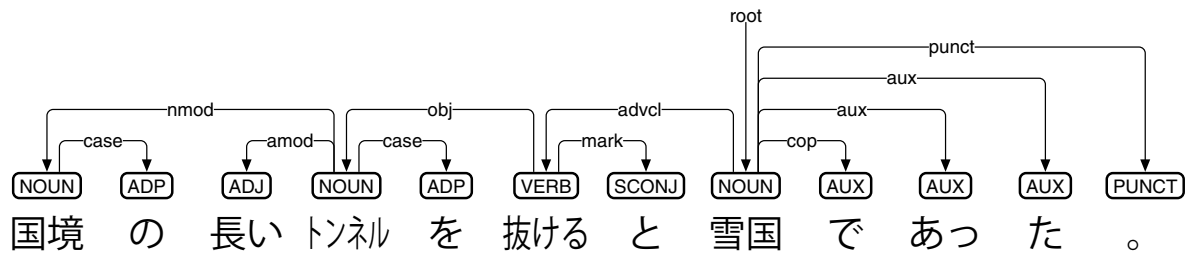


google/gemma-4-31b-it

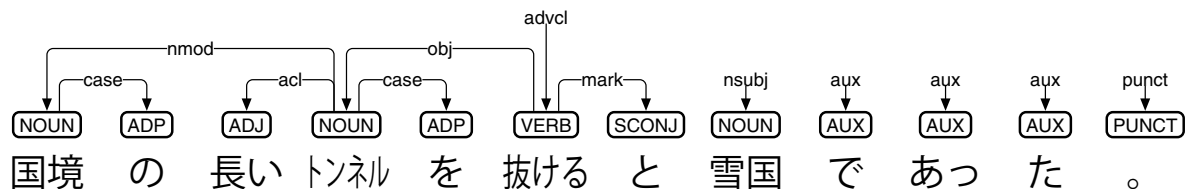


「抜ける」「雪国」のリンク元が「であった」になっている。

moonshotai/kimi-k2.6

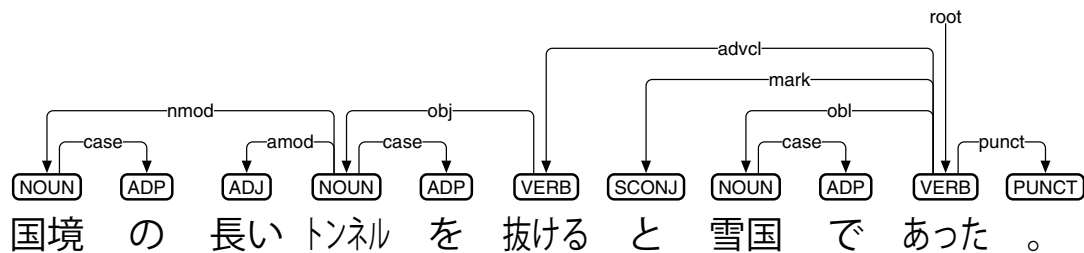


moonshotai/kimi-k2.7-code

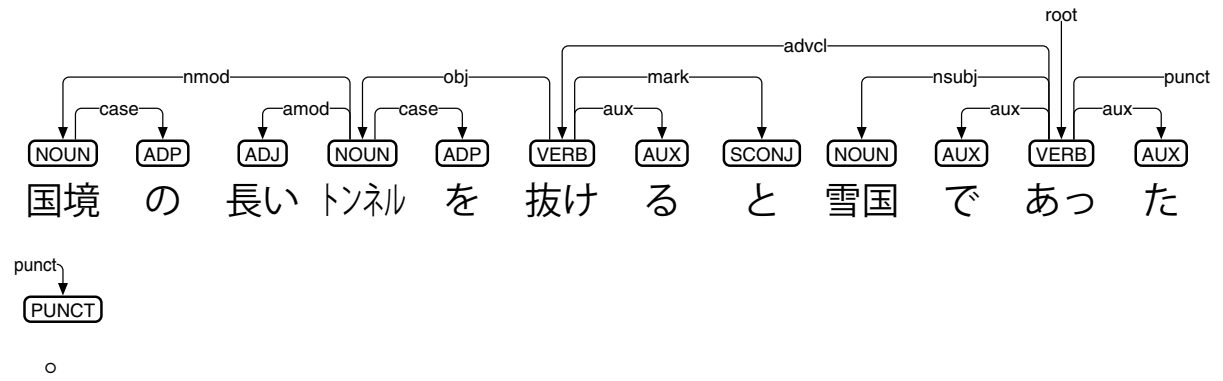


「抜ける」「雪国」「で」「あっ」「た」「。」のリンク元がいずれも「あった」になっており、しかも root が無い。

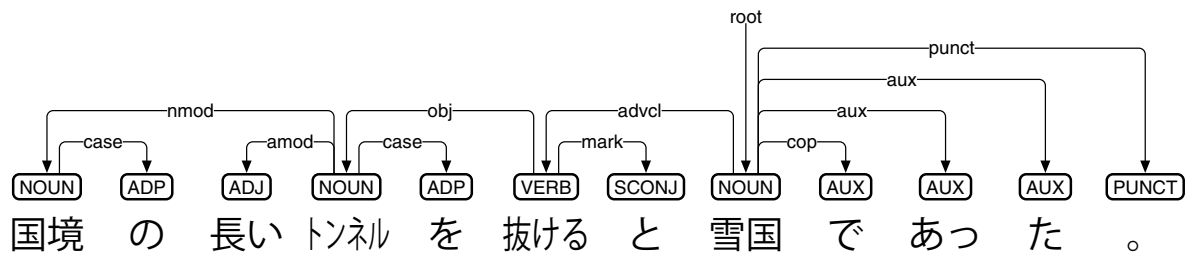
openai/gpt-oss-120b



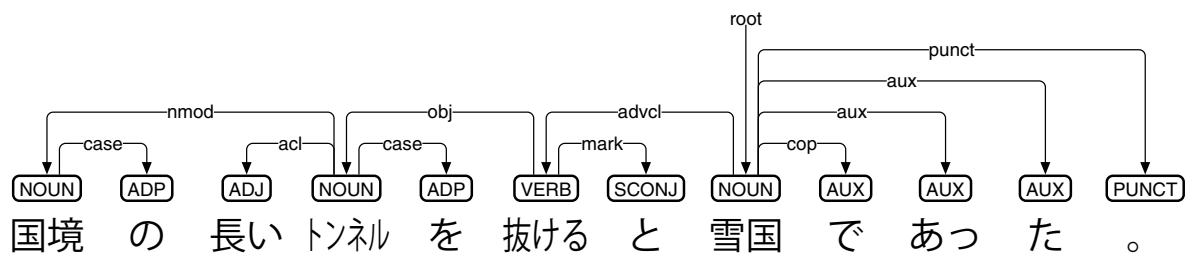
qwen/qwen3.6-27b



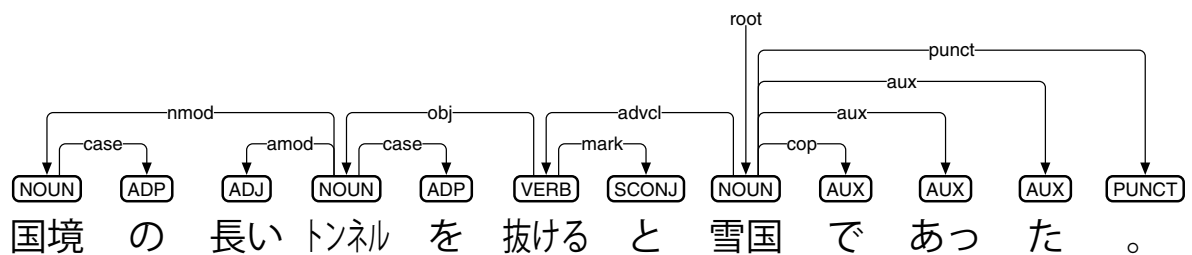
zai-org/glm-5.1



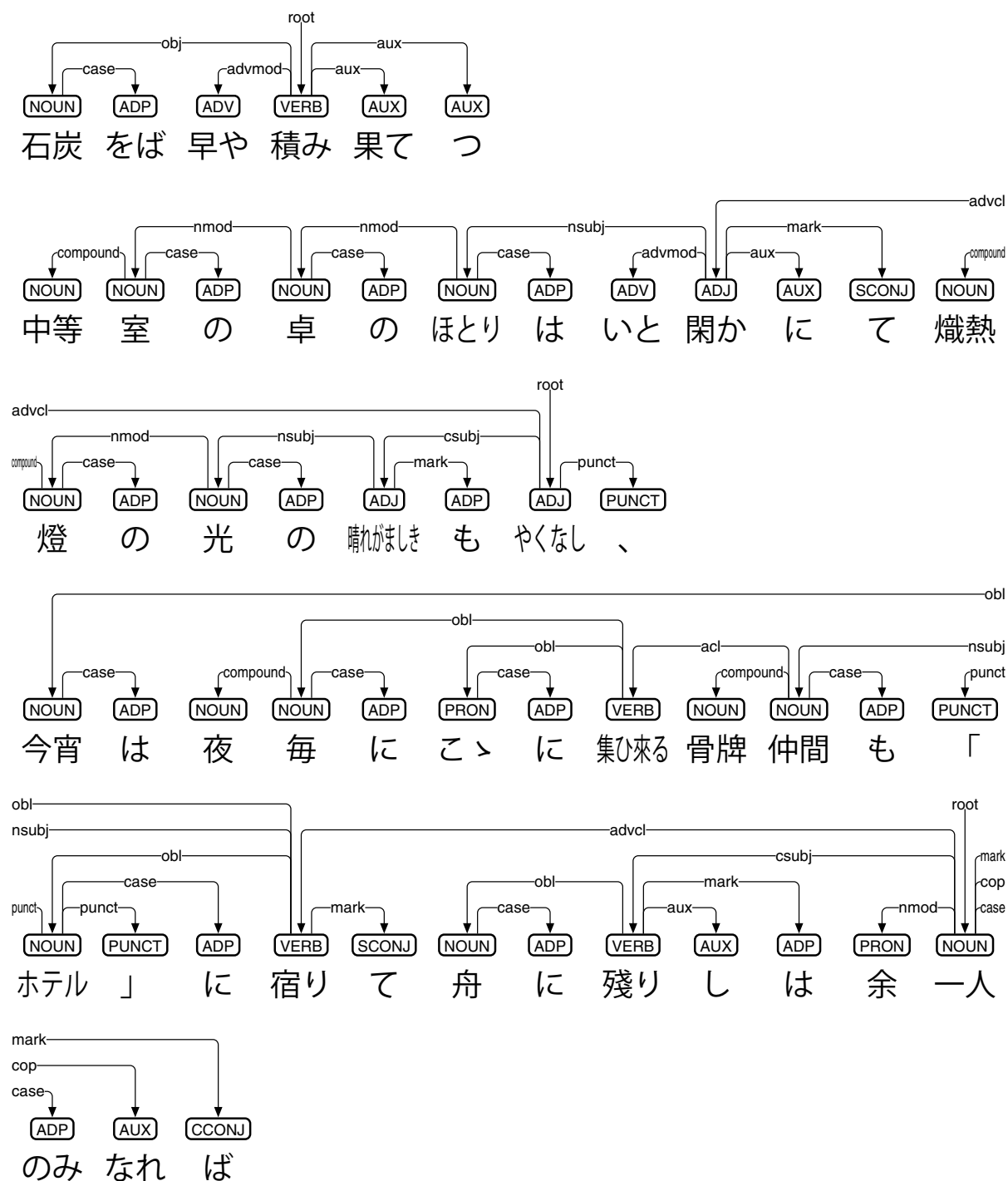
zai-org/glm-5.2



筆者の考える正解

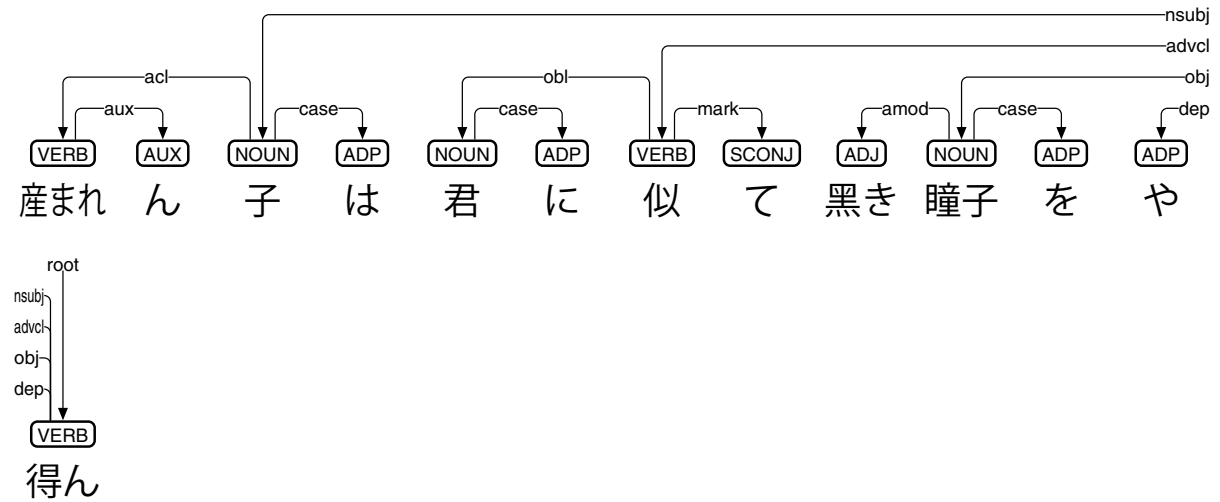


3.3 近代日本語 Few-Shot 係り受け解析



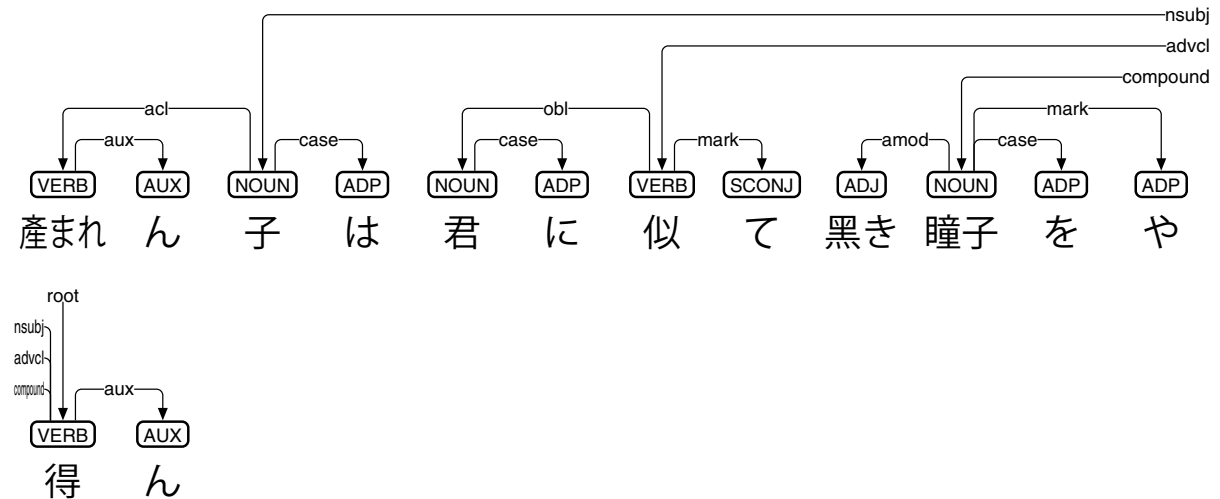
上記3つを例文として、「産まれん子は君に似て黒き瞳子をや得ん」の Three-Shot 係り受け解析に挑戦してみた。

deepseek-ai/deepseek-v4-flash

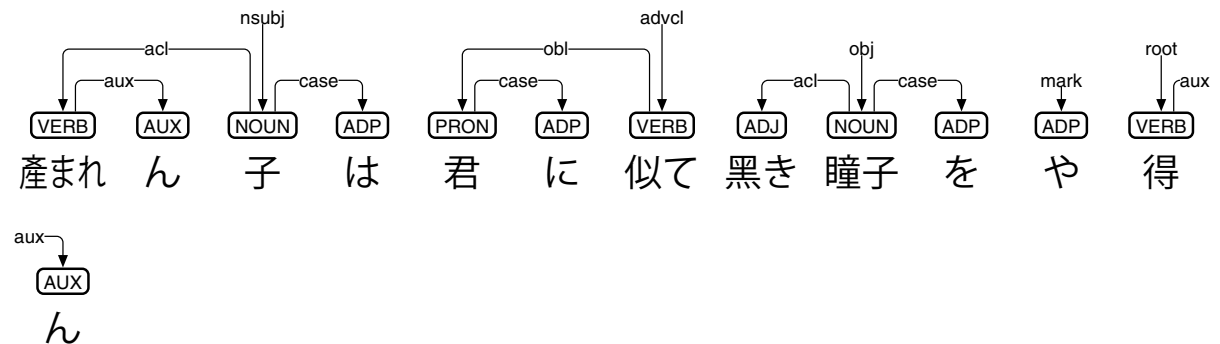


「産まれ」が「産まれ」に化けている。

deepseek-ai/deepseek-v4-pro

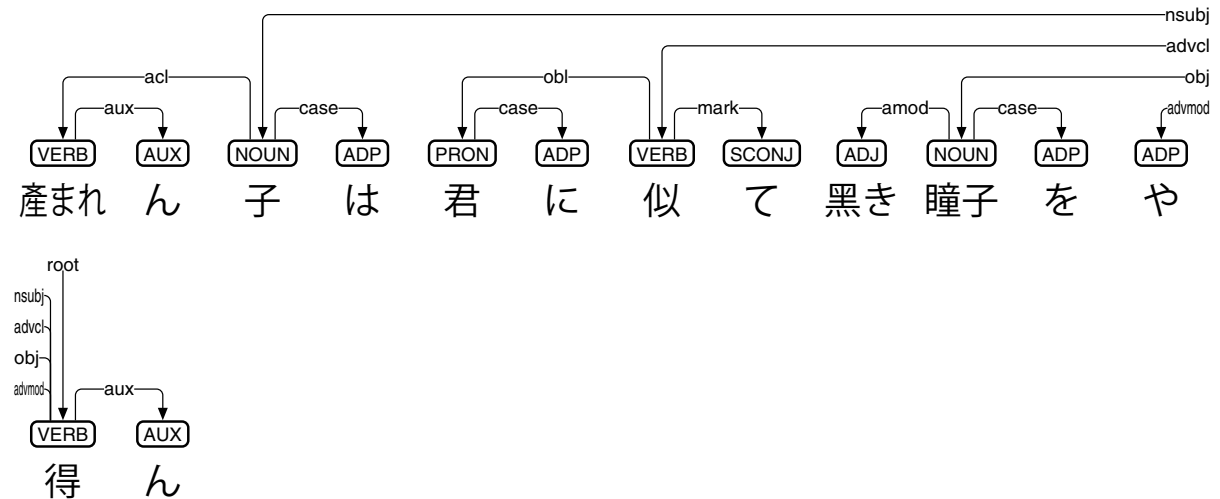


google/gemma-4-31b-it

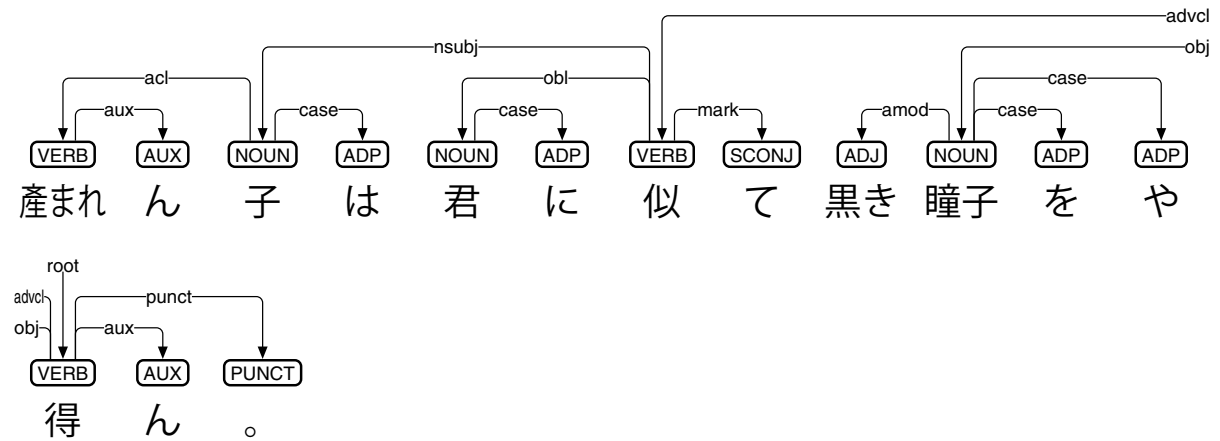


「子」「似て」「瞳子」「や」のリンク元が「得ん」になっている。

moonshotai/kimi-k2.6

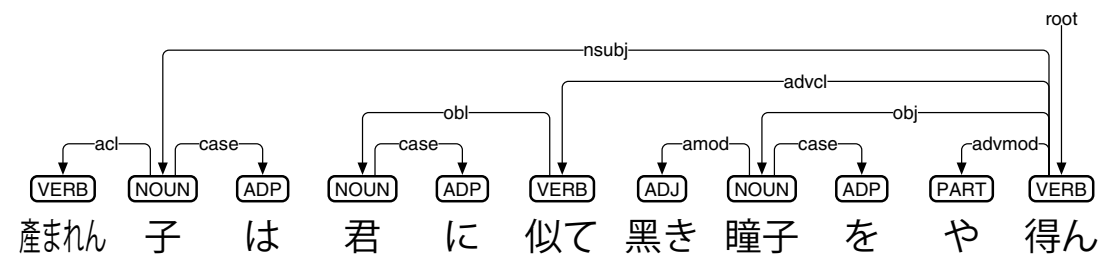


moonshotai/kimi-k2.7-code

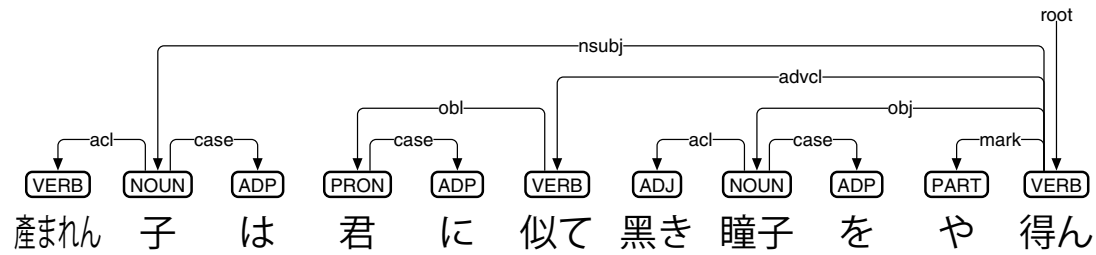


openai/gpt-oss-120b はリンク元が全く出力されておらず、フォーマットが壊れている。

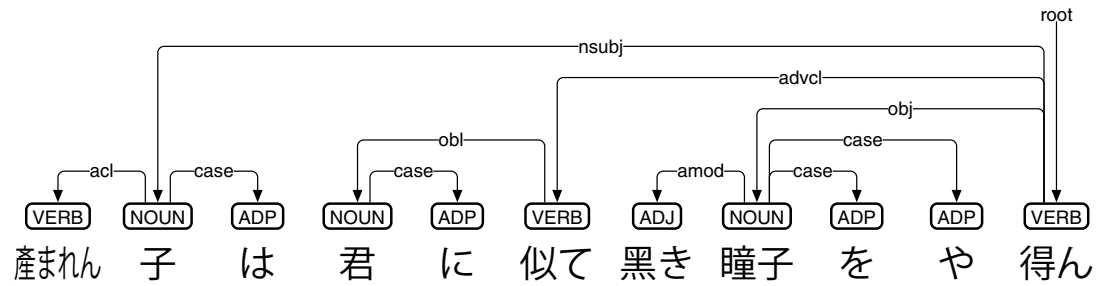
qwen/qwen3.6-27b



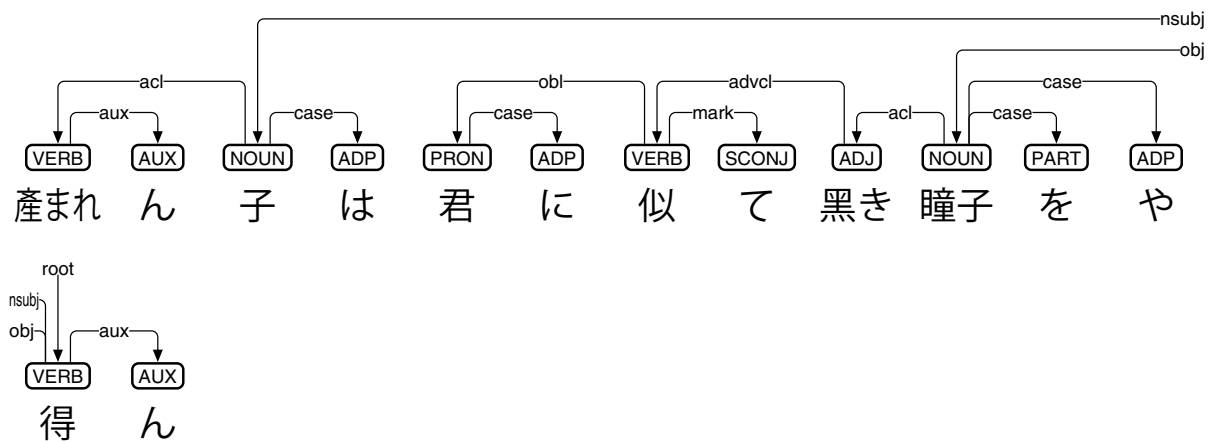
zai-org/glm-5.1



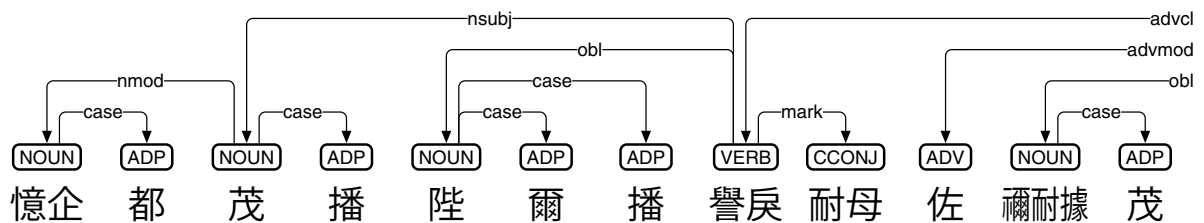
zai-org/glm-5.2

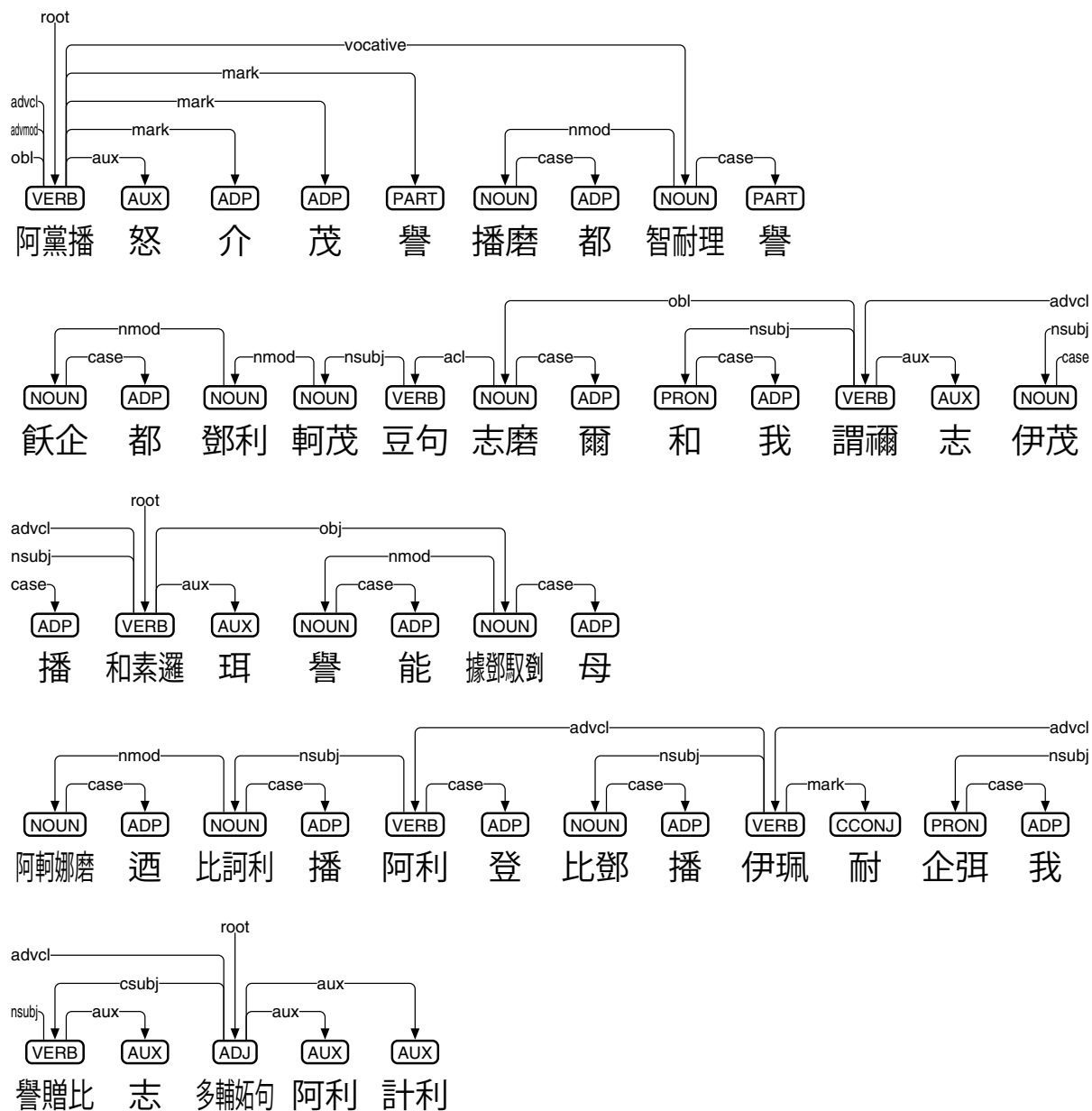


筆者の考える正解



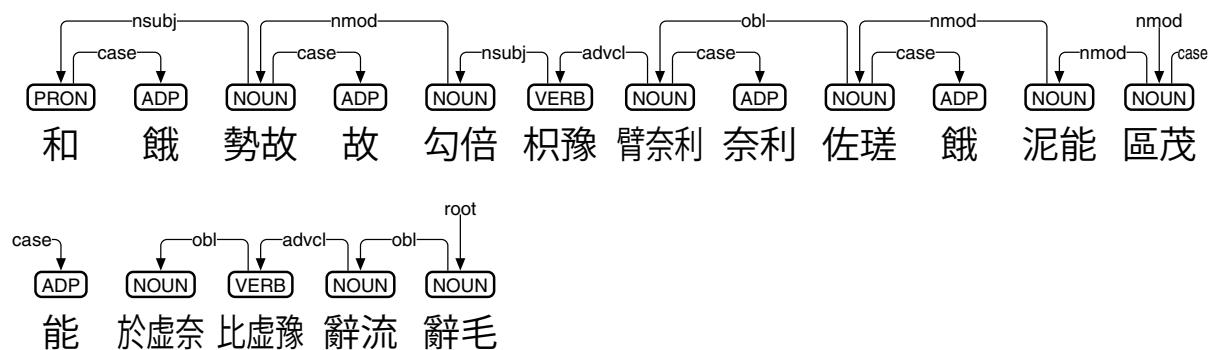
3.4 上代日本語 Few-Shot 係り受け解析





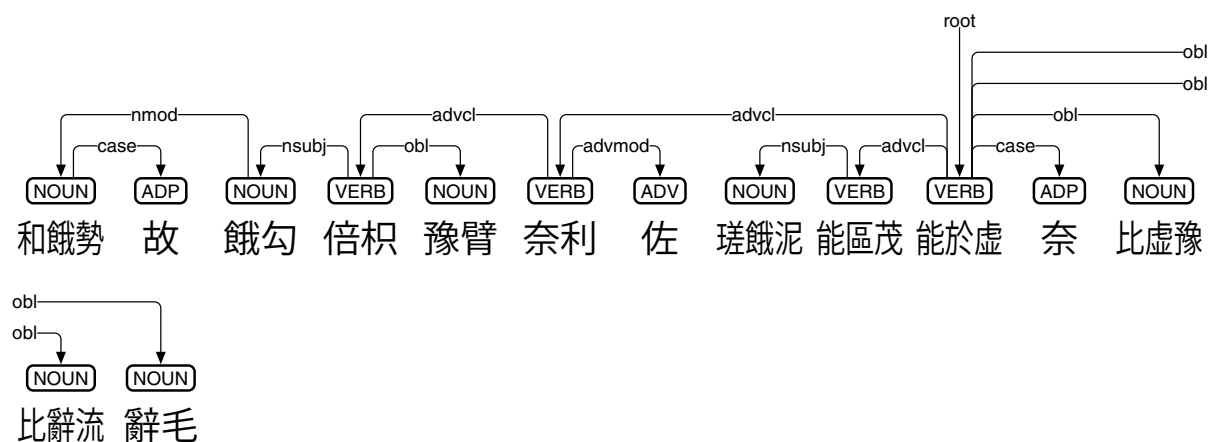
万葉仮名で書かれた上記3つを例文として「和餓勢故餓勾倍枳豫臂奈利佐瑳餓泥能區茂能於虚奈比虚豫比辭流辭毛」の Three-Shot 係り受け解析に挑戦してみた。

deepseek-ai/deepseek-v4-flash

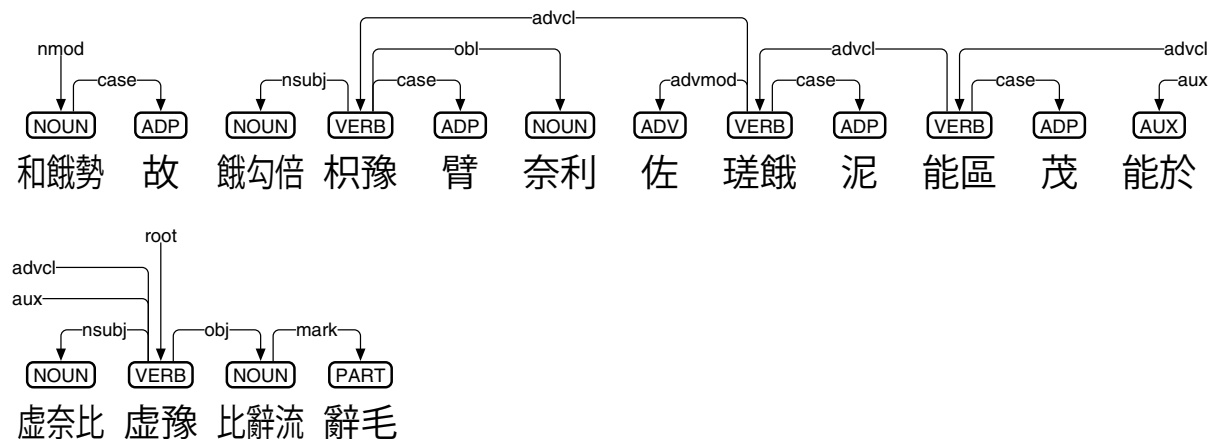


「勢故餓」が「勢故故」に化けている上に、「區茂」のリンク元が「能於虚奈」になっている。

deepseek-ai/deepseek-v4-pro

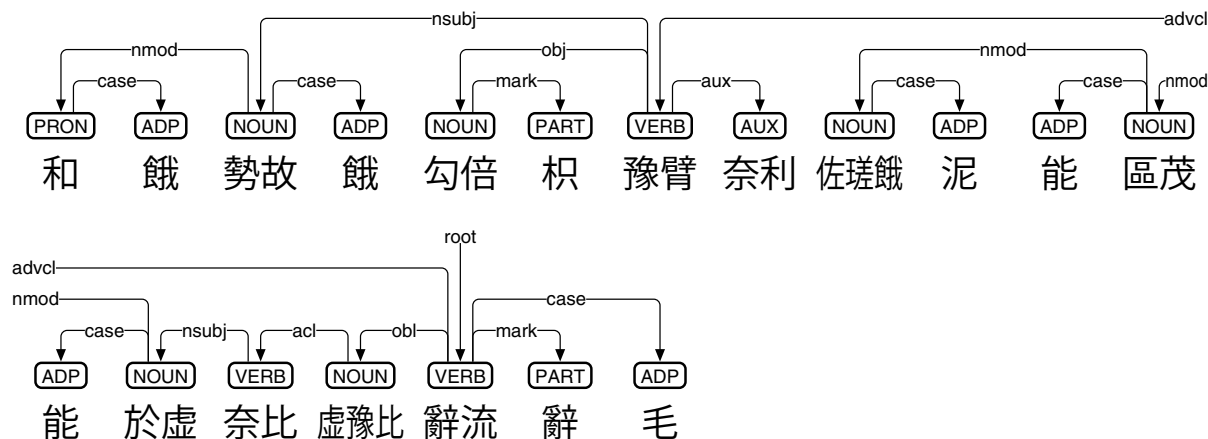


google/gemma-4-31b-it



「和餓勢」のリンク元が「故餓」になっている。

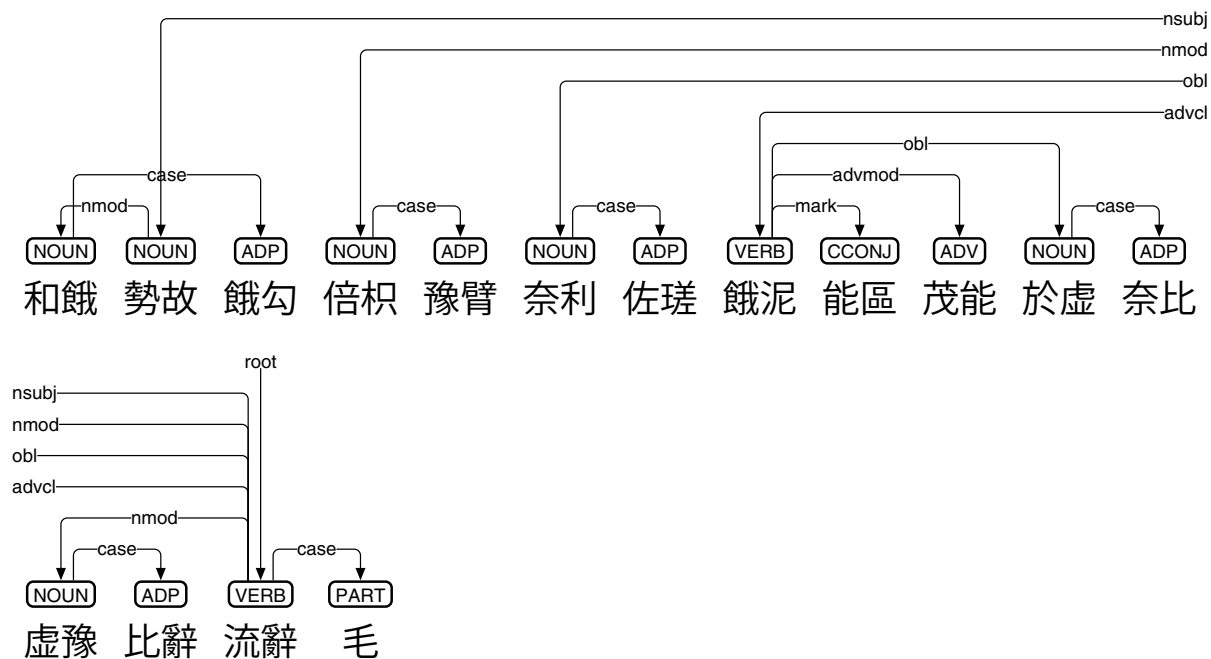
moonshotai/kimi-k2.6



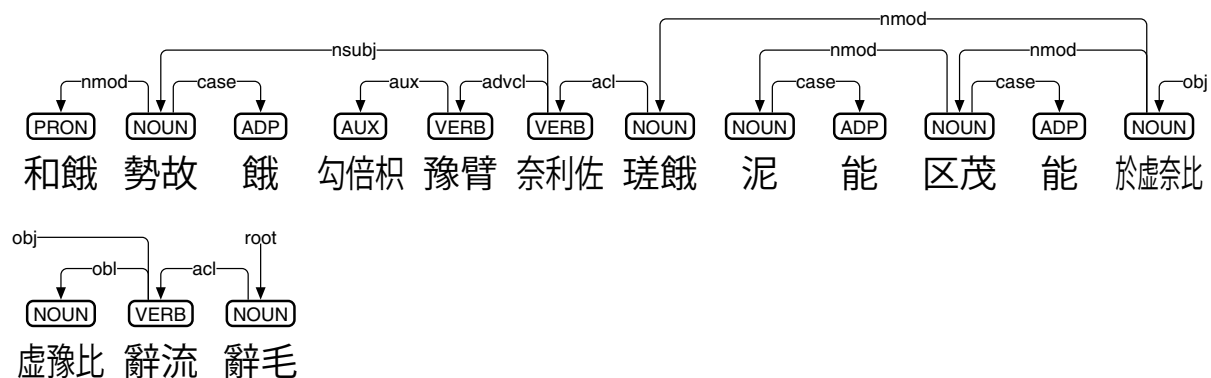
The diagram illustrates the syntactic structure of the sentence. It features a main clause structure with a subject NP (和 我), a verb phrase (勢故 餓 勾倍枳 豫), and a complement NP (腎奈利佐堯饒泥 能 區 茂 能 於). Below this, there is a prepositional phrase (虛奈比 虛豫比 辭流 辭 毛) which serves as an adjunct to the main clause. The diagram uses various grammatical relations to show how these elements are connected, such as 'nmod' for nominal modification, 'case' for case marking, 'nsubj' for nominal subject, 'acl' for adpositional clause, 'obl' for oblique, 'advcl' for adpositional clause, 'aux' for auxiliary, and 'mark' for marking. The characters are color-coded: red for pronouns (和, 我), green for verbs (勢故, 餓, 勾倍枳, 豫), blue for nouns (腎奈利佐堯饒泥, 能, 區, 茂, 能, 於), and grey for particles/auxiliaries (虛奈比, 虛豫比, 辭流, 辭, 毛).

– 18 –

qwen/qwen3.6-27b

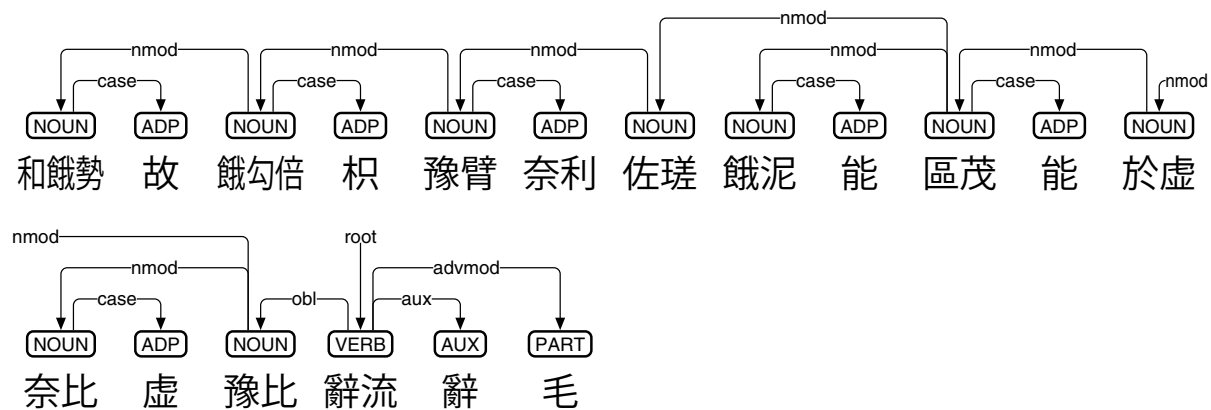


zai-org/glm-5.1

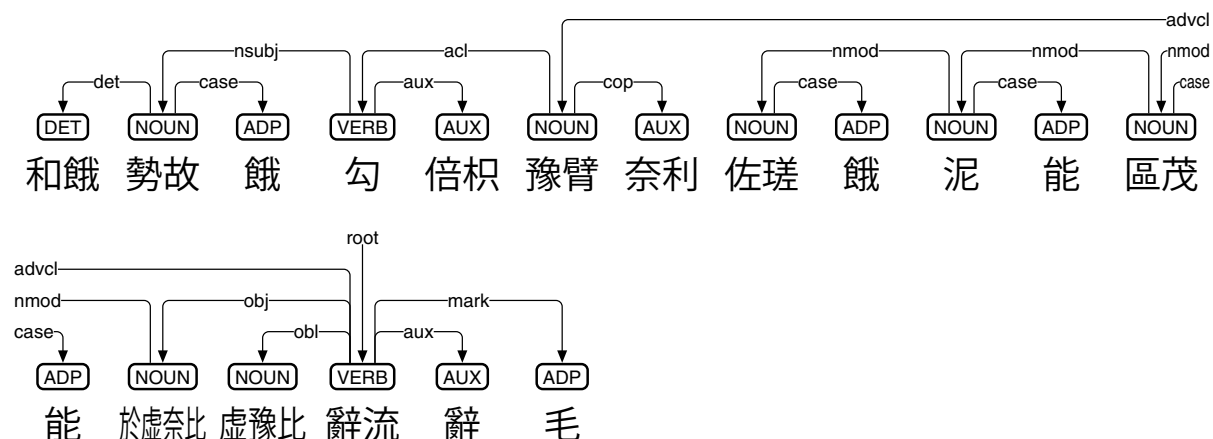


「區茂」が「区茂」に化けている。

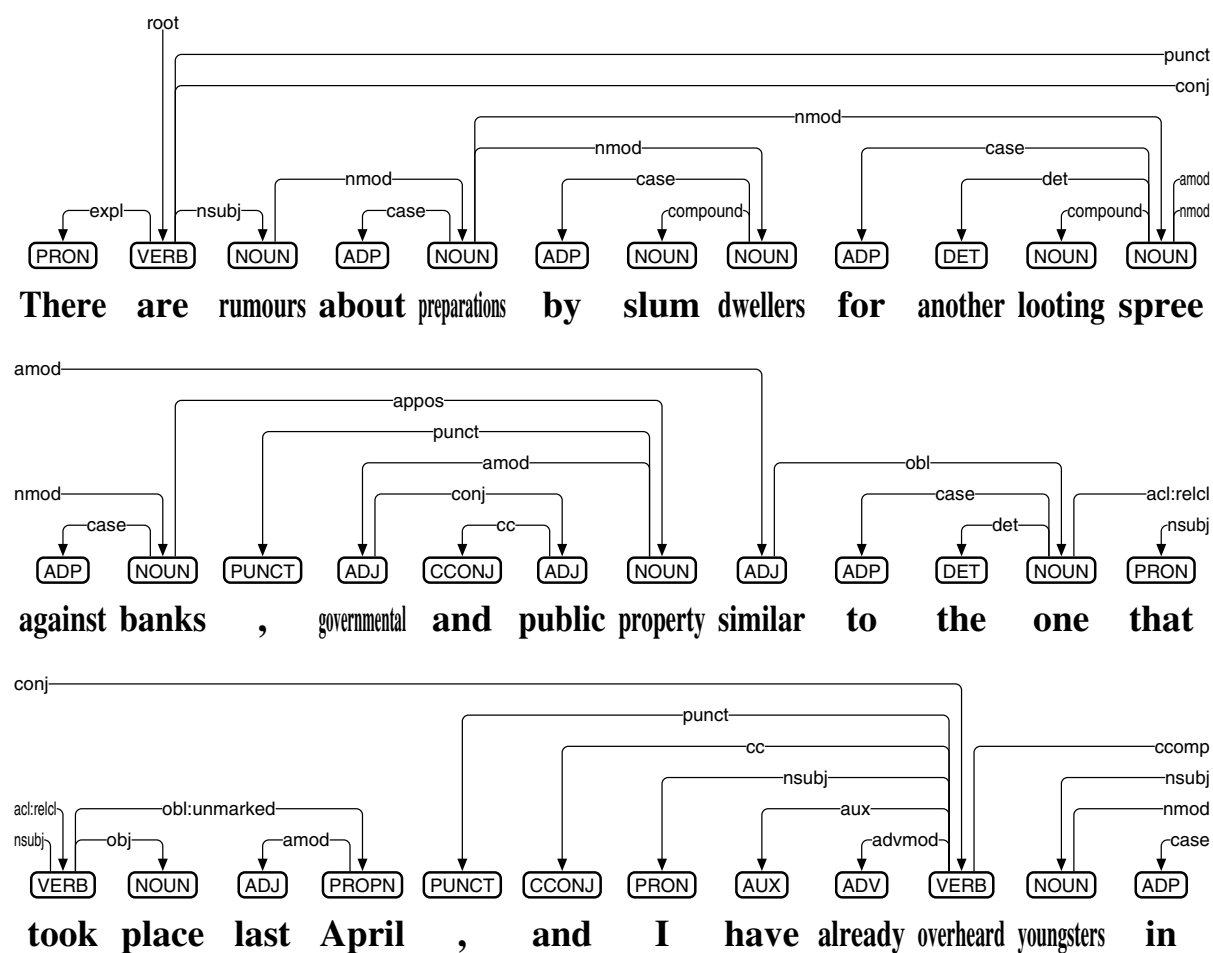
zai-org/glm-5.2

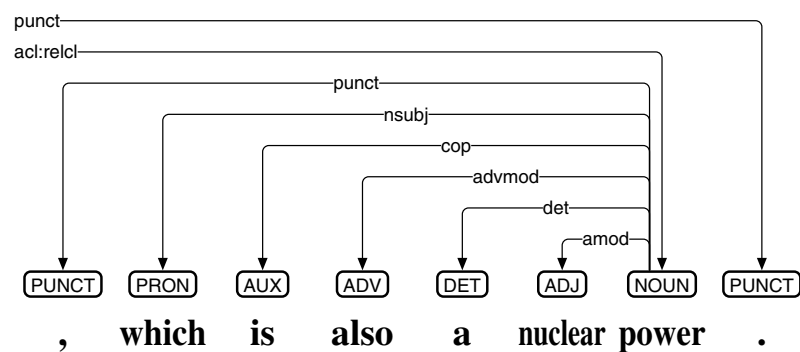
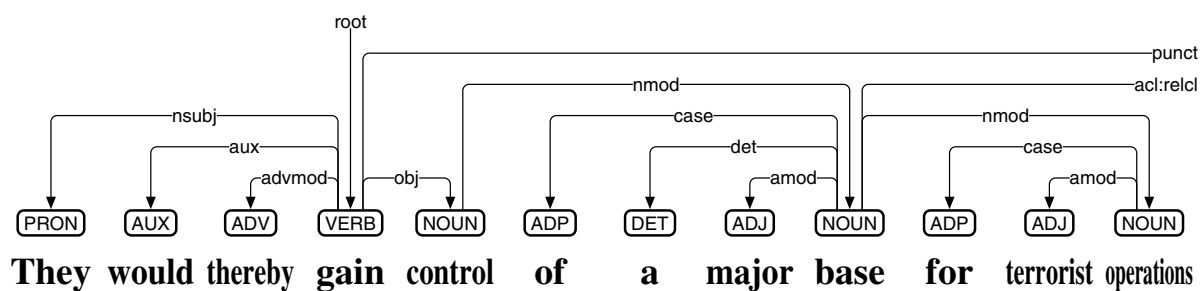
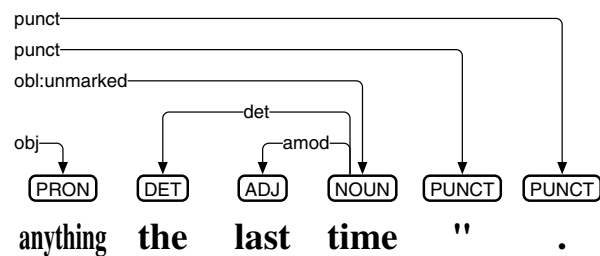
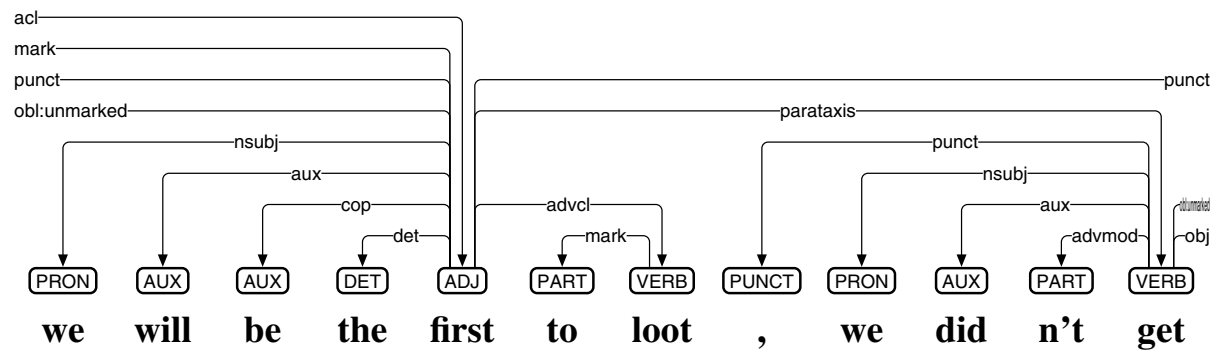
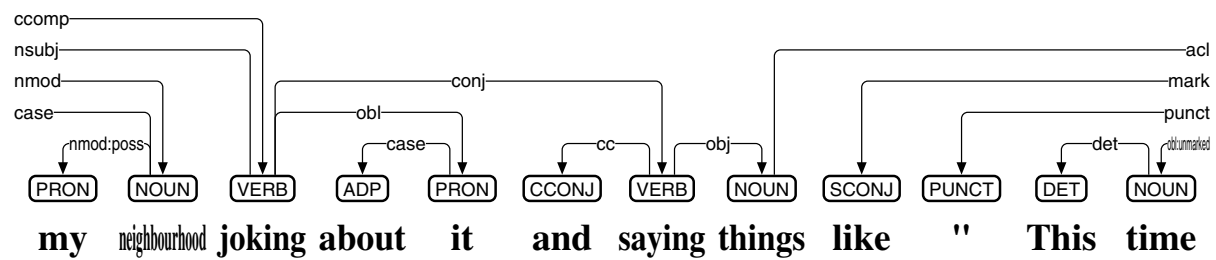


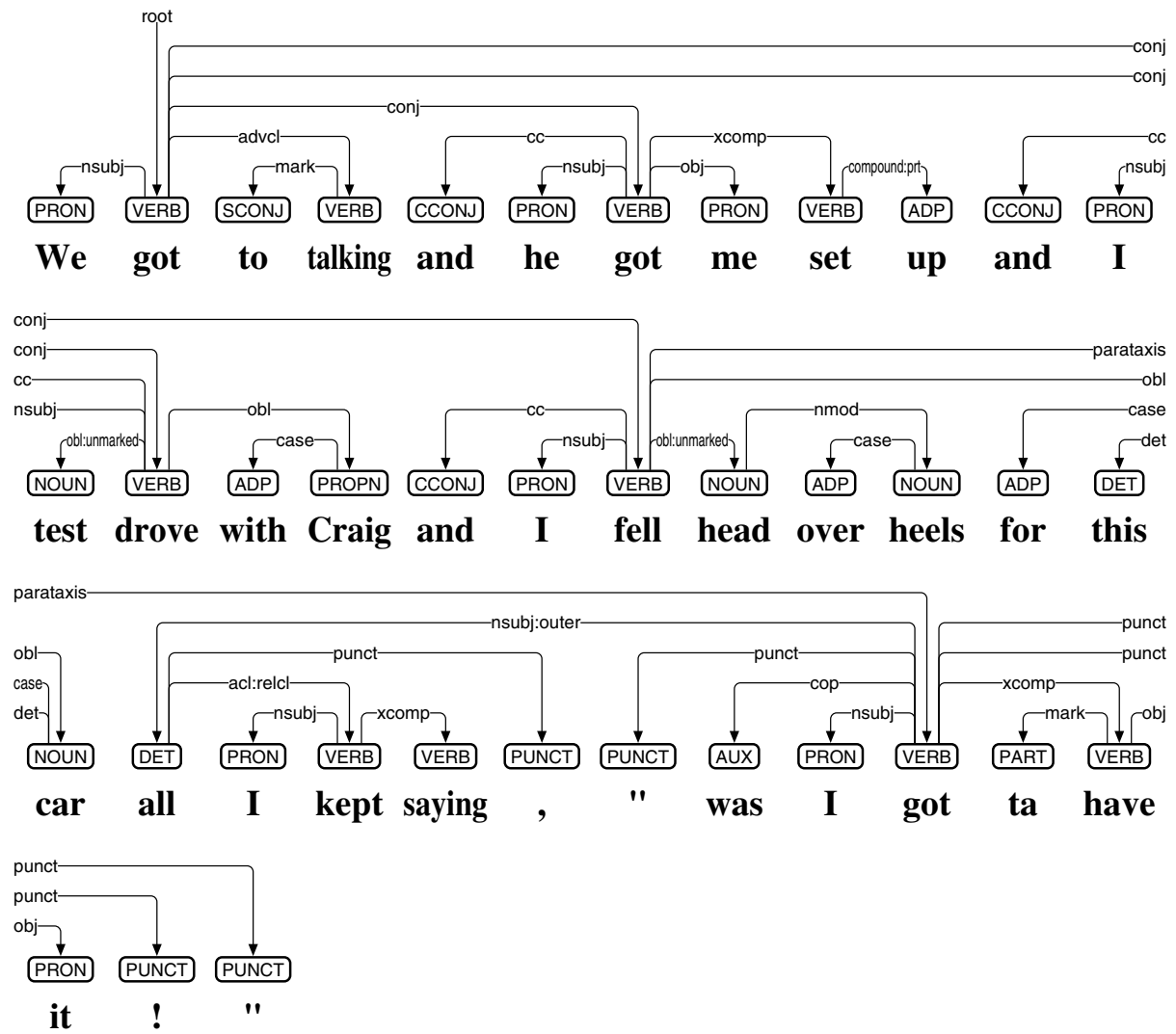
筆者の考える正解



3.5 英語 Few-Shot 係り受け解析

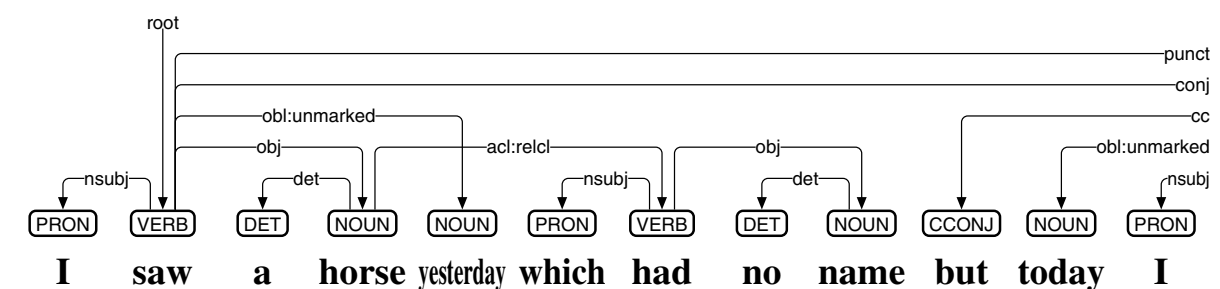


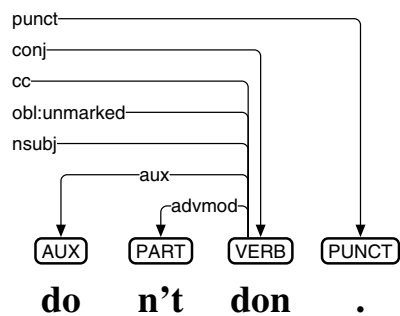




上記3つを例文として「I saw a horse yesterday which had no name but today I don't.」の Three-Shot 係り受け解析に挑戦してみた。

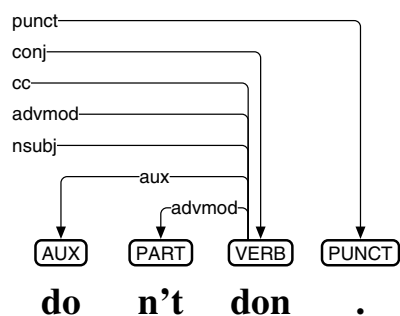
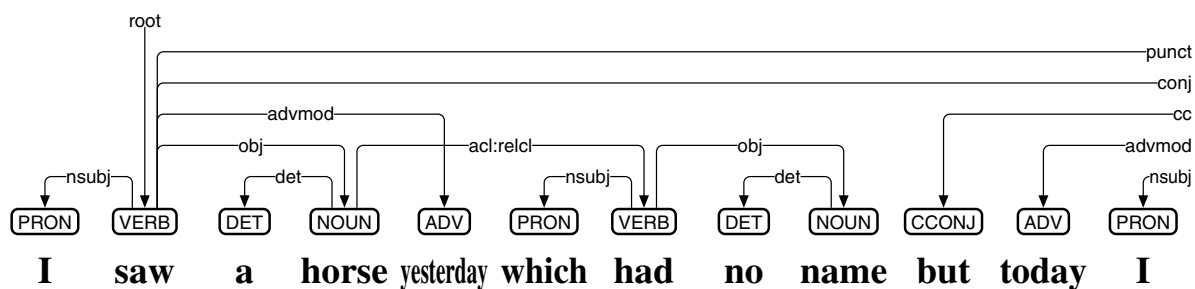
deepseek-ai/deepseek-v4-flash





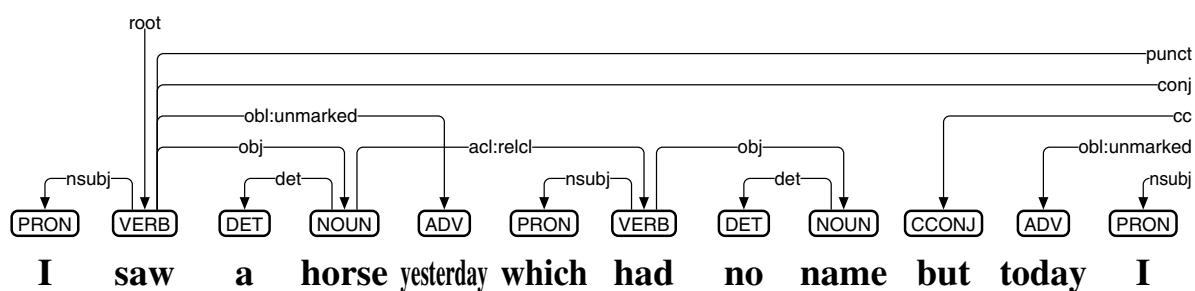
「don't」の後に「don」が追加されてしまっている。

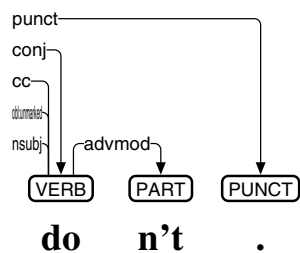
deepseek-ai/deepseek-v4-pro



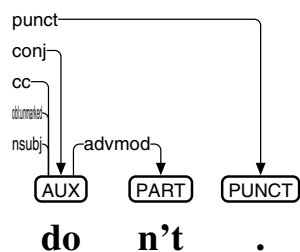
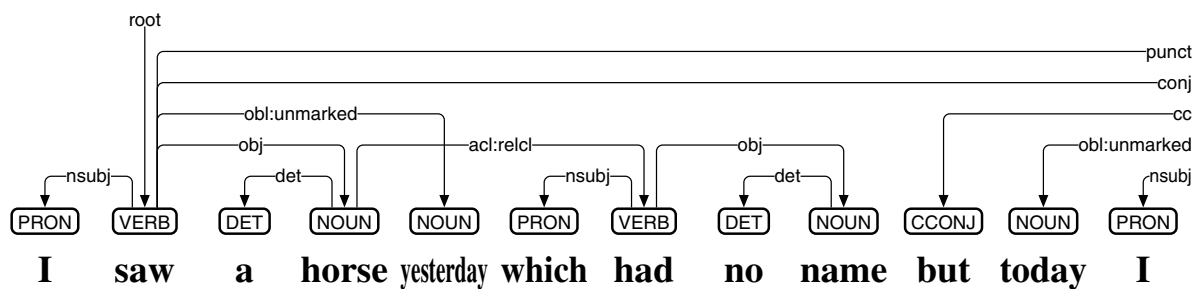
「don't」の後に「don」が追加されてしまっている。

google/gemma-4-31b-it

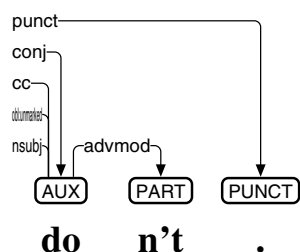
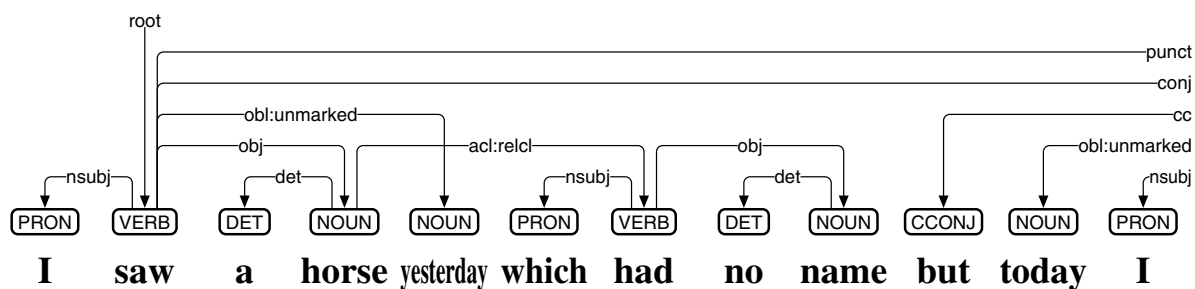




moonshotai/kimi-k2.6



moonshotai/kimi-k2.7-code



openai/gpt-oss-120b はフォーマットが壊れている。

The diagram illustrates the syntactic structure of the sentence "I saw a horse yesterday which had no name but today I do n't .". The words are arranged in two rows, with arrows indicating grammatical relationships between them. The first row contains the words: I, saw, a, horse, yesterday, which, had, no, name, but, today, I. The second row contains the words: do, n't, . (period). The diagram shows how these words are grouped into phrases and how they relate to each other grammatically. For example, "I" is the subject of "saw", "a" is the determiner for "horse", "yesterday" is the time modifier for "saw", "which" is the relative pronoun for "horse", "had" is the verb for "which", "no" is the determiner for "name", "but" is the conjunction, "today" is the time modifier for "do n't", and "I" is the subject of "do n't".

The diagram illustrates the syntactic structure of the sentence "I saw a horse yesterday which had no name but today I do n't .". The root node branches into several main components. The first part, "I saw a horse yesterday", is a clause where "I" is the subject (nsubj), "saw" is the main verb (v), "a horse" is the object (obj), and "yesterday" is an adverbial modifier (advmod). The second part, "which had no name", is a relative clause where "which" is the subject (nsubj), "had" is the main verb (v), and "no name" is the object (obj). The third part, "but today I", is a conjunction (cc) with an adverbial modifier (advmod) "today" and a subject (nsubj) "I". The final part, "do n't .", is a verb phrase where "do" is the auxiliary verb (aux), "n't" is the negation particle (part), and "." is the punctuation (punct). The diagram uses a hierarchical tree structure with arrows indicating the parent-child relationships between the words and their syntactic categories.

Diagram illustrating the syntax tree structure for the sentence: "I saw a horse yesterday which had no name but today I".

The root node branches into:

- punct** (punctuation)
- conj** (conjunction)
- cc** (coordinator)
- obl:tmod** (oblique modifier)
- obj** (object)
- acl:relcl** (adpositional clause relative clause)
- obl:tmod** (oblique modifier)

The **obl:tmod** nodes further branch into:

- nsubj** (nominal subject)
- verb** (verb)

The **obj** node branches into:

- det** (determiner)
- noun** (noun)

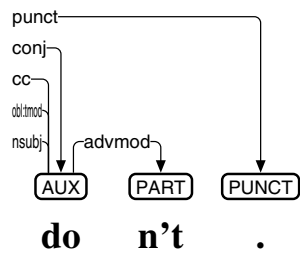
The **acl:relcl** node branches into:

- nsubj** (nominal subject)
- verb** (verb)

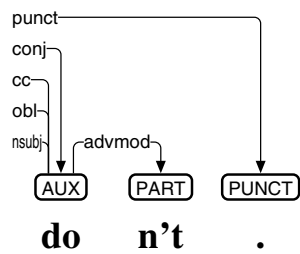
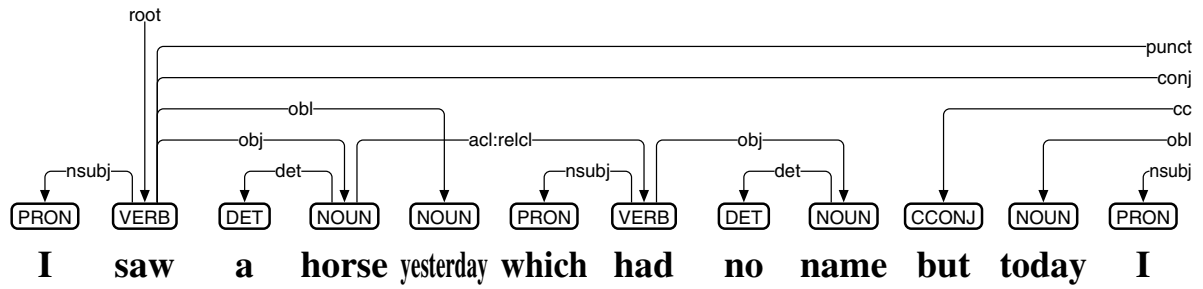
The **obl:tmod** node branches into:

- conj** (conjunction)
- noun** (noun)
- pron** (pronoun)

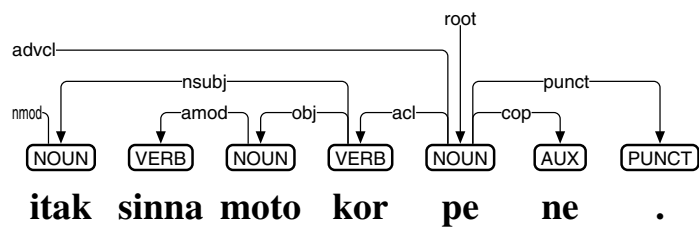
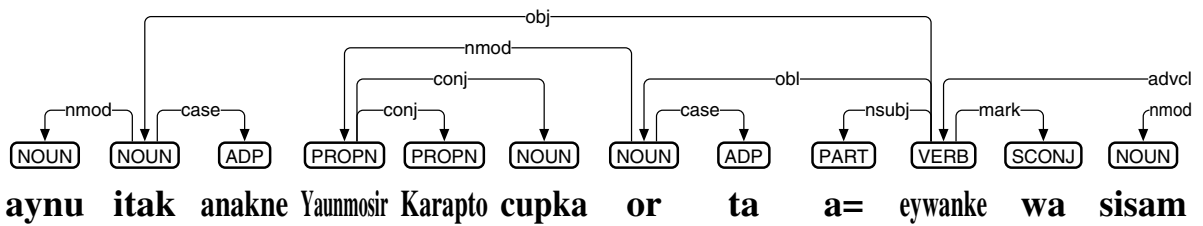
The corresponding words in the sentence are: "I", "saw", "a", "horse", "yesterday", "which", "had", "no", "name", "but", "today", "I".

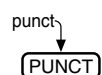
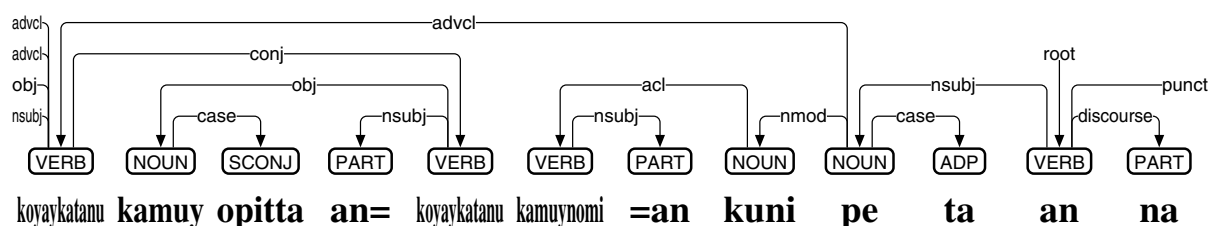
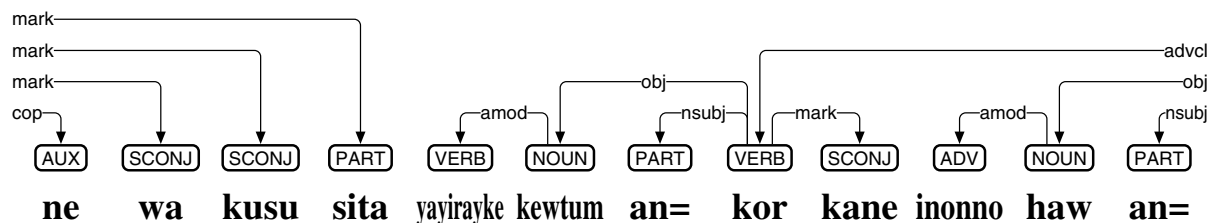
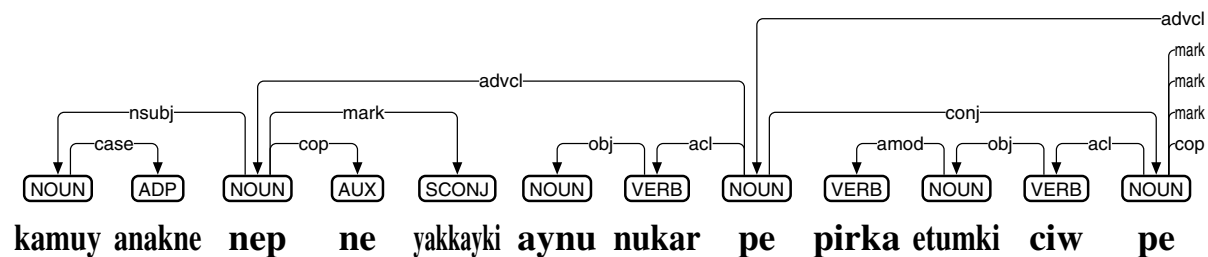


筆者の考える正解

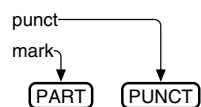
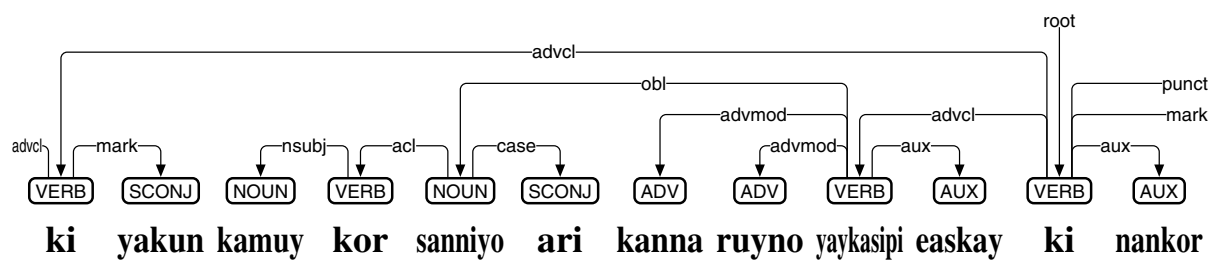
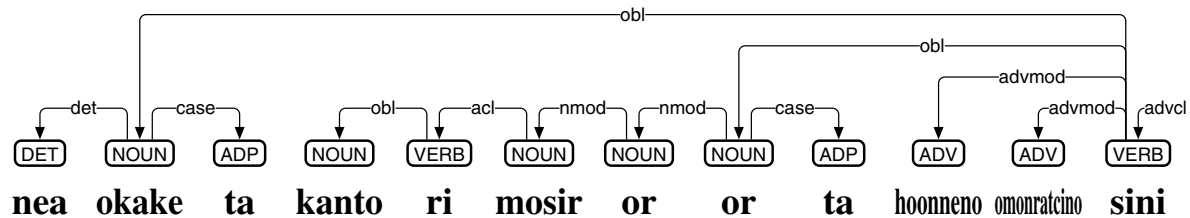


3.6 アイヌ語 Few-Shot 係り受け解析





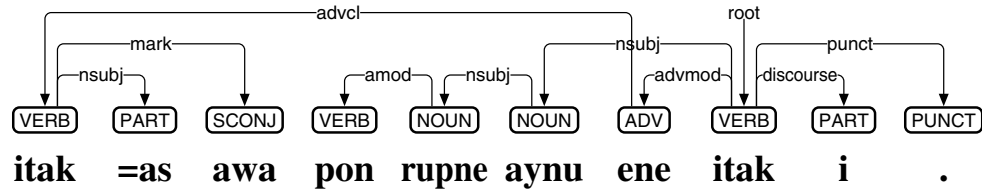
.



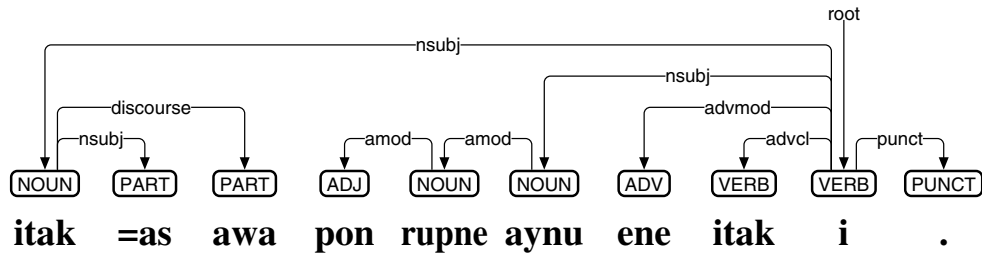
.

上記3つを例文として「itak=as awa pon rupne aynu ene itak i.」のThree-Shot 係り受け解析に挑戦してみた。

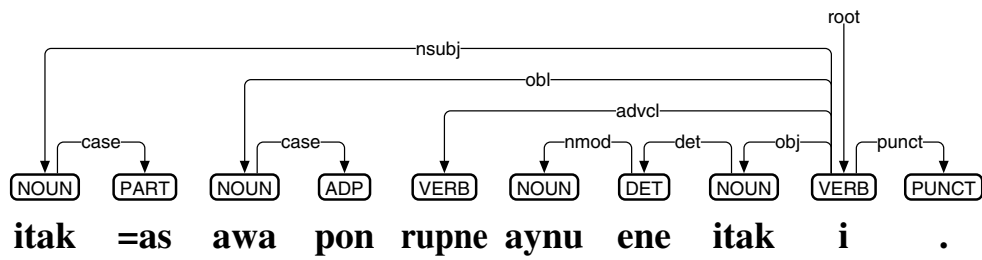
deepseek-ai/deepseek-v4-flash



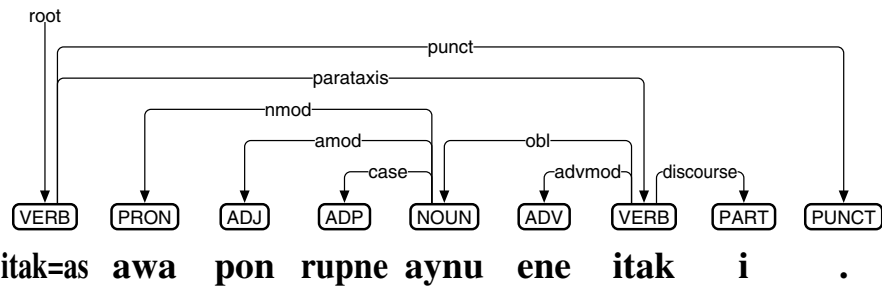
deepseek-ai/deepseek-v4-pro



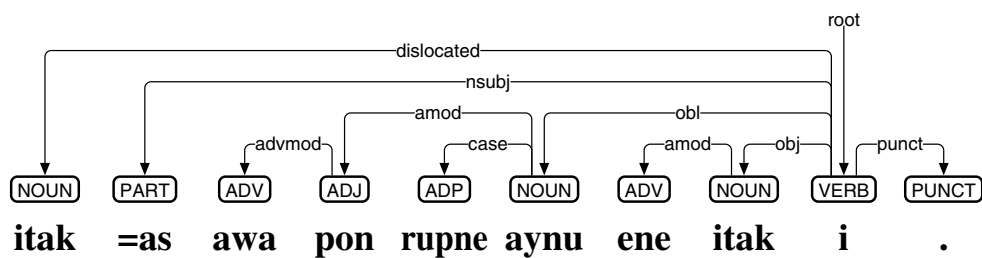
google/gemma-4-31b-it



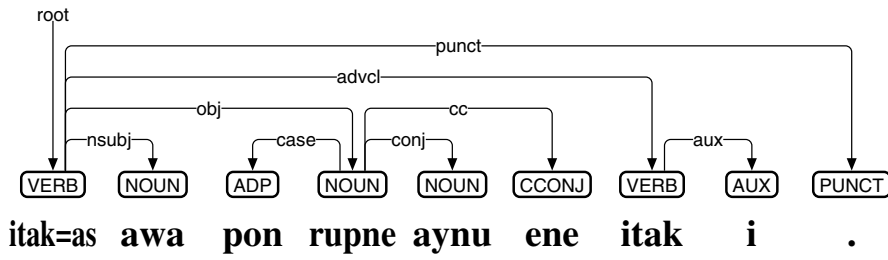
moonshotai/kimi-k2.6



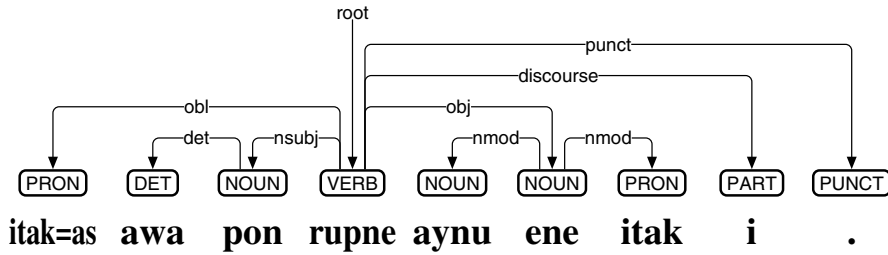
moonshotai/kimi-k2.7-code



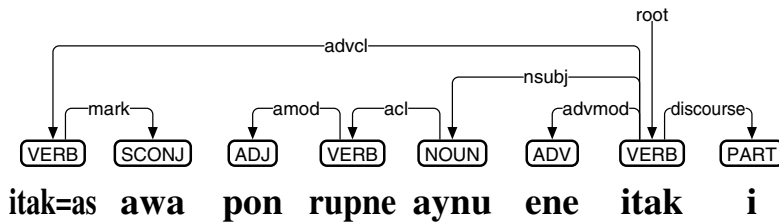
openai/gpt-oss-120b



qwen/qwen3.6-27b

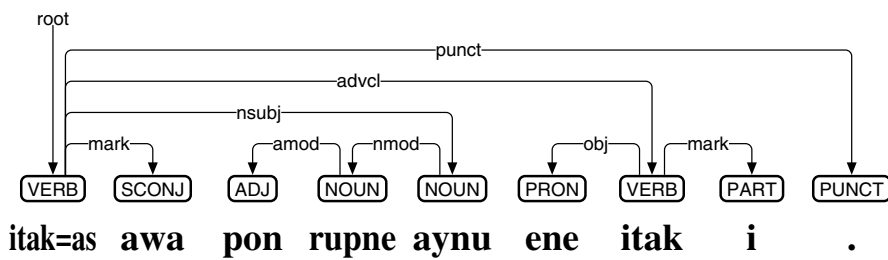


zai-org/glm-5.1

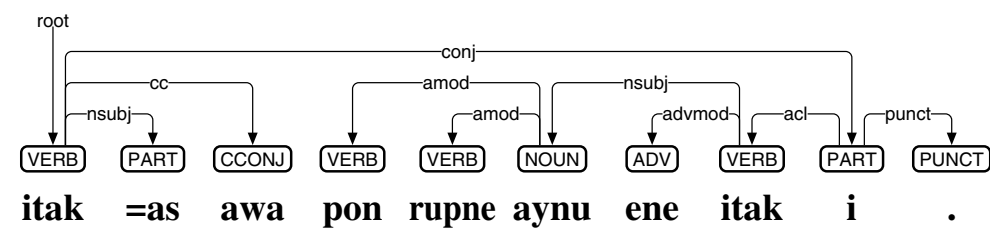


文末のピリオドが脱落している。

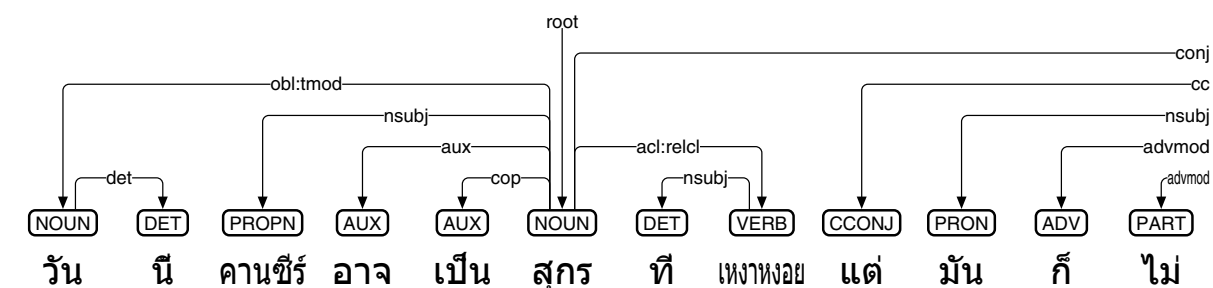
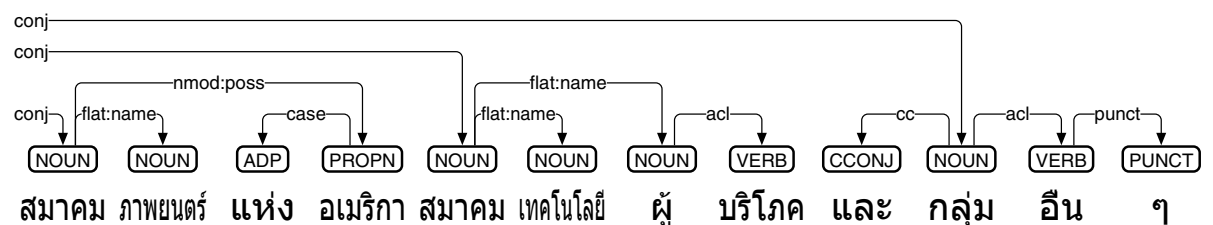
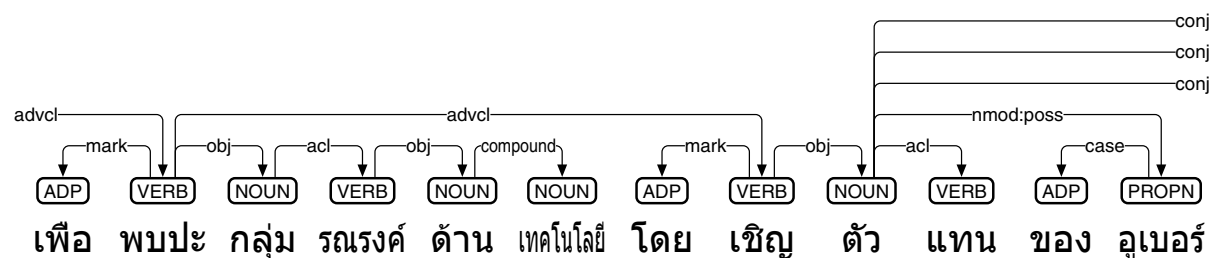
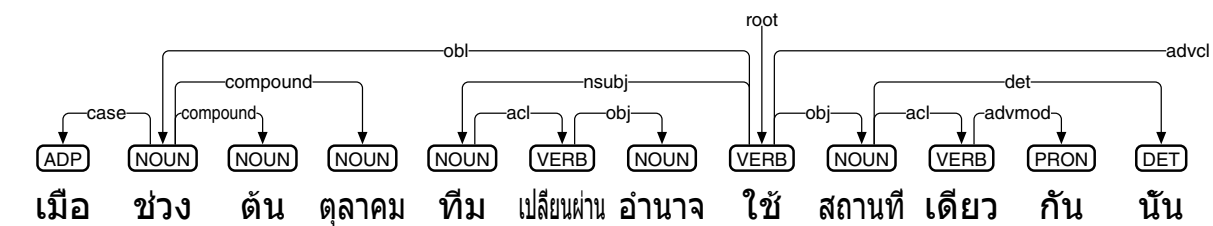
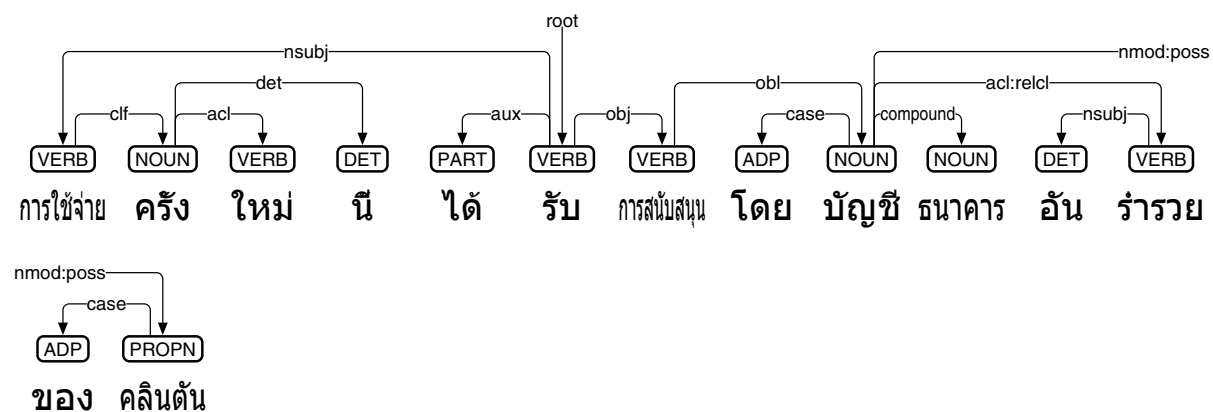
zai-org/glm-5.2

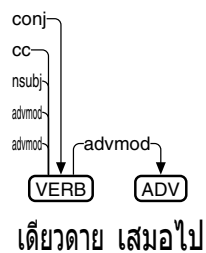


筆者の考える正解



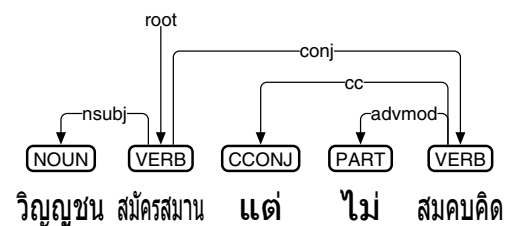
3.7 タイ語 Few-Shot 係り受け解析



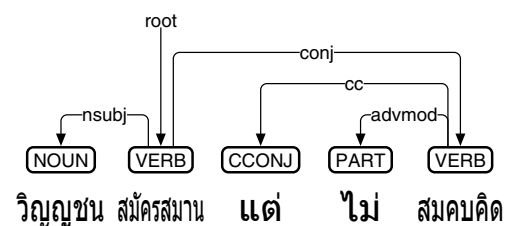


上記3つを例文として「วิญญาณสมควรसानแต่ไม่สมคบคิด」の Three-Shot 係り受け解析に挑戦してみた。

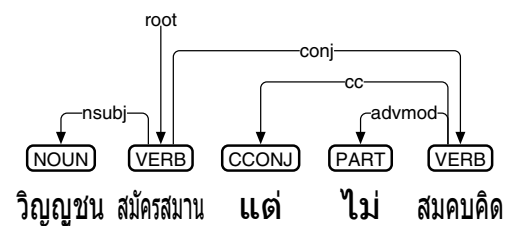
deepseek-ai/deepseek-v4-flash



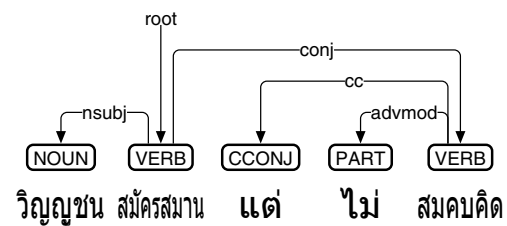
deepseek-ai/deepseek-v4-pro



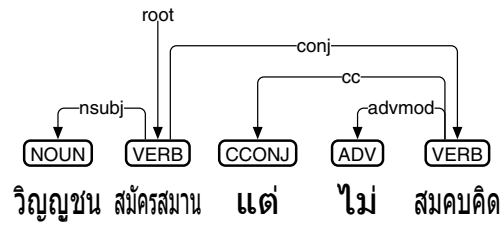
google/gemma-4-31b-it



moonshotai/kimi-k2.6

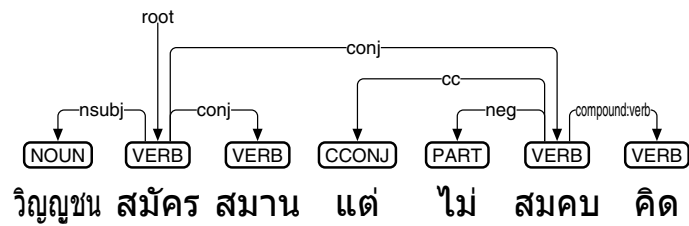


moonshotai/kimi-k2.7-code

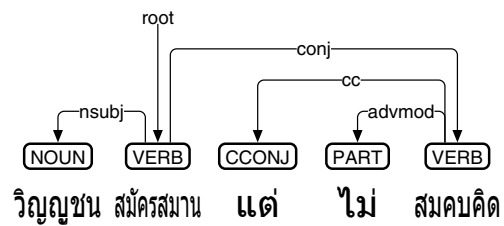


openai/gpt-oss-120b はリンク元が全く出力されておらず、フォーマットが壊れている。

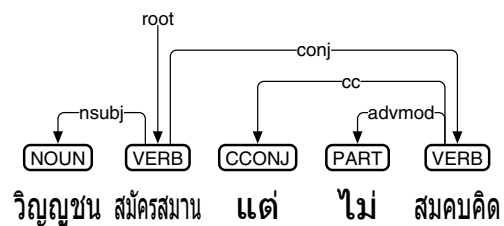
qwen/qwen3.6-27b



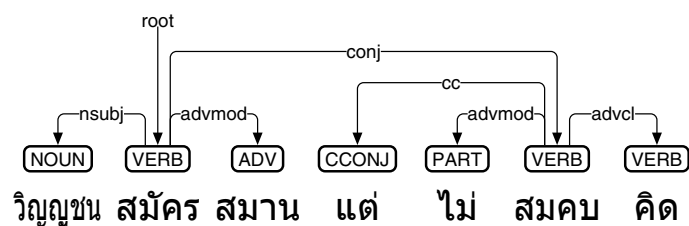
zai-org/glm-5.1



zai-org/glm-5.2



筆者の考える正解



4 考察

前章の係り受け解析の結果を、CoNLL 2018 [9] の UPOS/UAS/LAS で評価した。評価結果を以下に示す。「×」は評価不能を意味する。

	kimi-k2.6	qwen3.6-27b	glm-5.2
古典中国語	100/100/90	100/100/100	100/100/90
現代日本語	100/100/100	80/32/32	100/100/92
近代日本語	93/86/79	40/32/24	48/40/32
上代日本語	49/39/22	6/6/0	39/28/17
英語	100/100/100	100/93/87	100/100/100
アイヌ語	53/42/21	42/0/0	42/21/0
タイ語	50/0/0	86/100/57	50/0/0

	deepseek-v4-flash	deepseek-v4-pro	gemma4-31b-it	kimi-k2.7-code	gpt-oss-120b	glm-5.1
古典中国語	×	100/80/80	100/100/100	100/100/100	×	100/100/70
現代日本語	×	92/50/50	×	×	78/52/43	100/100/100
近代日本語	×	86/93/71	×	×	×	48/32/32
上代日本語	×	6/0/0	×	×	×	×
英語	×	×	80/100/100	100/100/100	×	87/100/87
アイヌ語	80/50/30	50/50/30	40/40/10	40/50/20	×	×
タイ語	50/0/0	50/0/0	50/0/0	33/0/0	×	50/0/0

全体をざっと見た限りでは kimi-k2.6 がダントツで、glm-5.2 と qwen3.6-27b が続いている。ただ、言語によって得手不得手があり、近代日本語は kimi-k2.6 が、タイ語は qwen3.6-27b が、かなりイイセンである。一方、その他のモデルは、出力文が化けたり、フォーマットが壊れたりすることがあるため、使用に耐えない。

ただし「係り受け解析したい文」を入れ替えた場合には、異なる結果が得られる可能性がある。同一語が複数出現するような文の扱いも問題である。また、本稿では、4つ組 (FORM・UPOS・DEPREL・リンク元 FORM) を一度に出力する手法を採用したが、むしろこれらを順に出力した方が、解析精度が上がるらしい [3, 4]。このあたりを、今後もう少し細かく考えていくべきだろう。

参考文献

- [1] Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, Daniel Zeman: Universal Dependencies, Computational Linguistics, Vol.47, No.2 (June 2021), pp.255-308.
- [2] 安岡孝一: GPT 系言語モデルによる国語研長単位係り受け解析, 人文科学とコンピュータシンポジウム「じんもんこん 2024」論文集 (2024 年 12 月), pp.83-90.
- [3] Hiroshi Matsuda, Chunpeng Ma, Masayuki Asahara: Step-by-step Instructions and a Simple Tabular Output Format Improve the Dependency Parsing Accuracy of LLMs, Proceedings of the 18th International Conference on Parsing Technologies (August 2025), pp.11-19.
- [4] 松田寛, 馬春鵬, 浅原正幸: 段階的指示と表形式出力を用いた大規模言語モデルによる依存構造解析, 自然言語処理, Vol.33, No.2 (2026 年 6 月), pp.920-956.

- [5] Lucien Tesnière: *Éléments de Syntaxe Structurale*, Paris: C. Klincksieck (1959).
- [6] Igor A. Mel'čuk: *Dependency Syntax: Theory and Practice*, New York: State University of New York Press (1988).
- [7] Joakim Nivre: *Towards a Universal Grammar for Natural Language Processing*, CICLing 2015: 16th International Conference on Intelligent Text Processing and Computational Linguistics (April 2015), pp.3-16.
- [8] 安岡孝一: Universal Dependencies にもとづく多言語係り受け可視化ツール deplacy, 人文科学とコンピュータシンポジウム「じんもんこん 2020」論文集 (2020 年 12 月), pp.95-100.
- [9] Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov: *CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Proceedings of the CoNLL 2018 Shared Task (October 2018), pp.1-21.