ZA Seminar

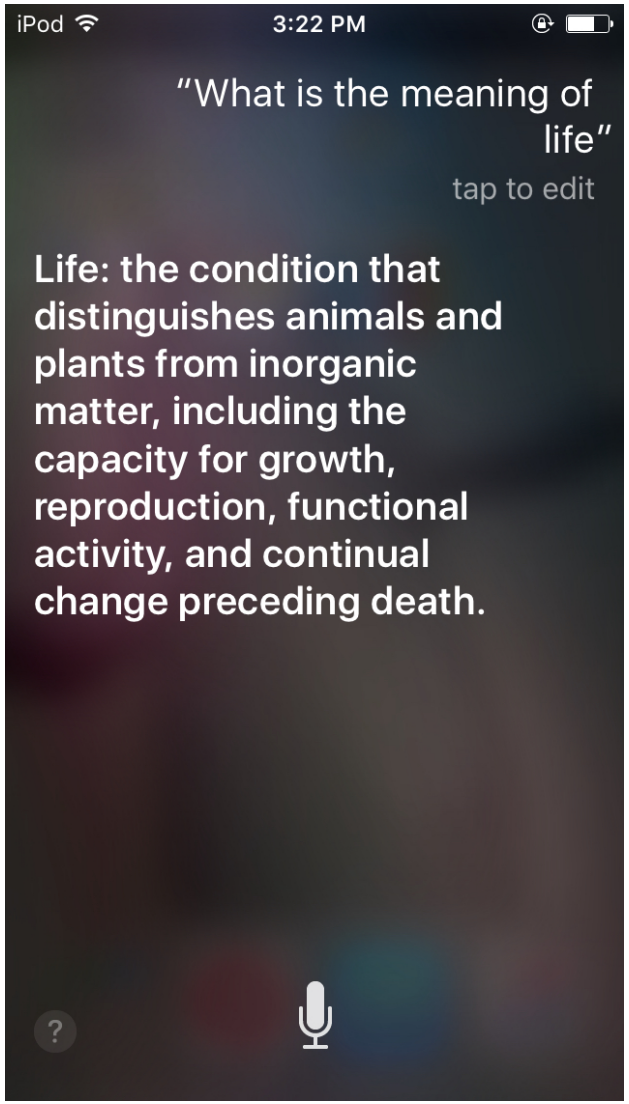
Lê Hoàng Trung

le.hg.trung@gmail.com

An approach for QA using dependency parsing

Tp. Hồ Chí Minh, Tháng 03/2018

Question Answering



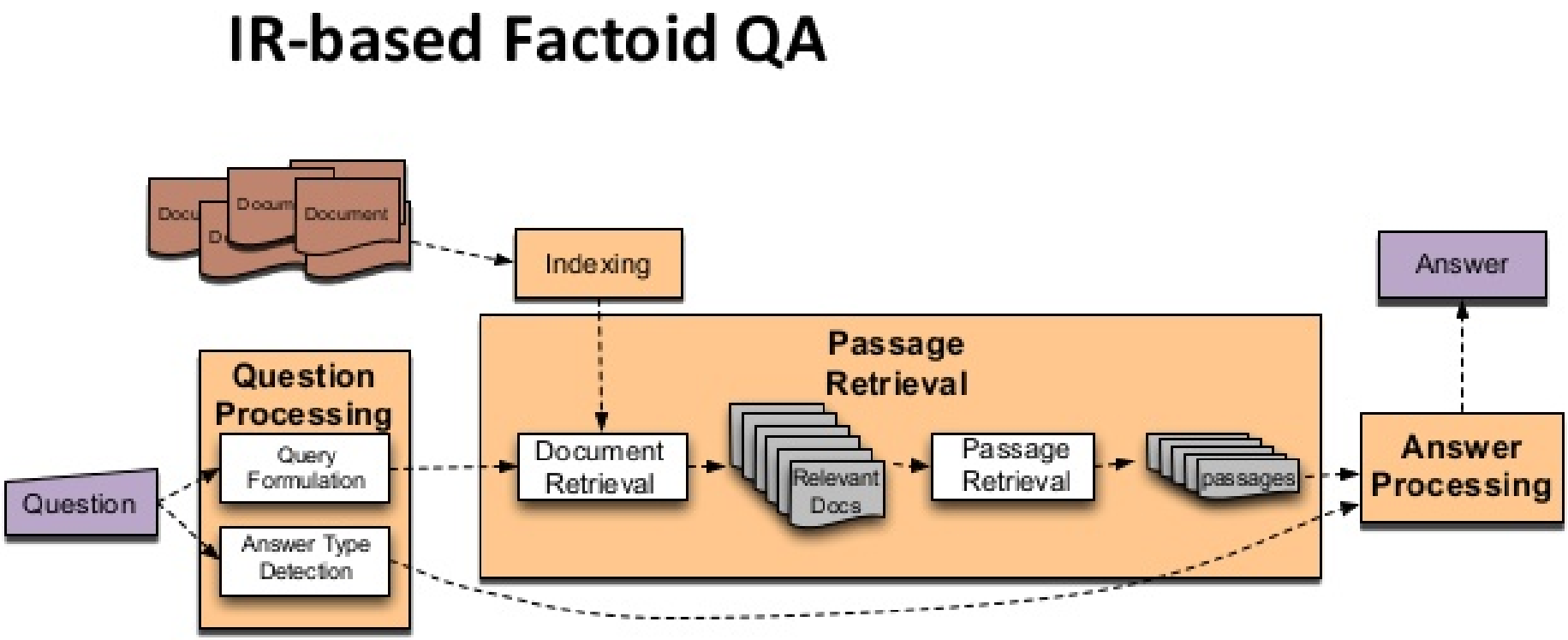
Type of Questions

- Factoid questions
  - Who is Putin?
  - How much water should a person drink a day?
  - Who wrote the book forrest gump was based on?
  - Does C++11, 14, 17 or 20 introduce a standard constant for pi?
- Non-factoid questions
  - What is the meaning of life?
  - Does writing matter a lot in research?
  - How can I give out my telephone number to my neighbors without implying anything?

Approaches for QA

- Information Retrieval approaches
  - Answer the question based on textual data.
  - Given a question, IR systems find out the paragraph that answer for it.

Information Retrieval approaches



Information Retrieval approaches

Q: *Which US state capital has the largest population?*

- **Answer Type:** city
- **Query:** US state capital, largest, population
- **state captital**

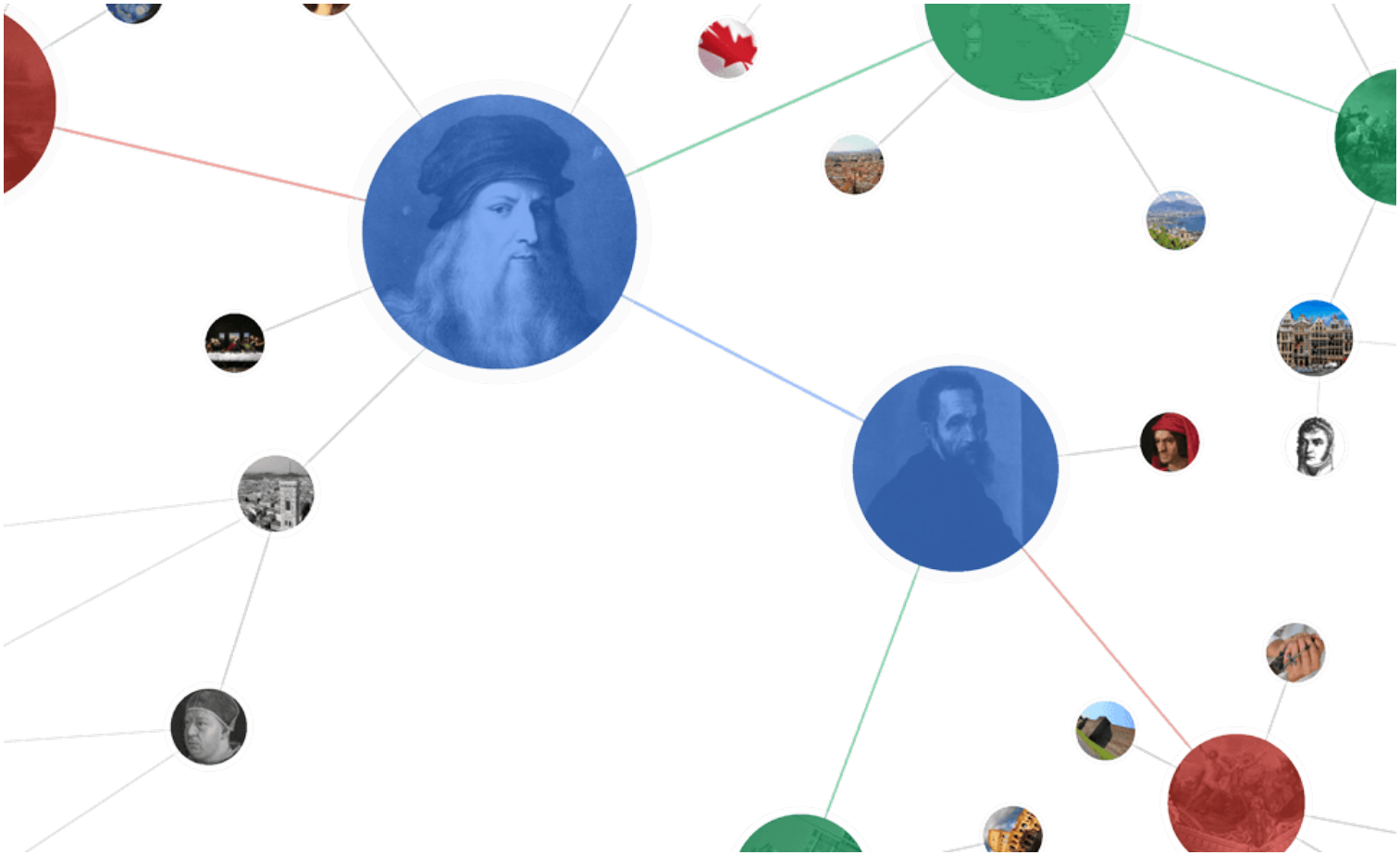
Approaches for QA

- Knowledge-based and Hybrid approaches
  - We convert the question into a semantic representation (kind of SQL queries).
  - Ex: *"What countries has population over 100 milions?"* $\Rightarrow$ *SELECT country.name FROM country WHERE country.population* $> 10^9$

Approaches for QA

- Information Retrieval approaches
  - Based on retriving and reranking documents.
  - Easier to implement.
  - Not precise comparing to KB
  - We don't understand the meanning of the questions.
- Knowledge-based and Hybrid approaches
  - Base on exact queries over a structured database.
  - Hard to implement (of course).
  - Precise
  - Help us understand the **semantic meaning** of the questions.

Knowledge graph



Knowledge graph

Mã Siêu

Mã Siêu, tự Mạnh Khởi 孟起, là một vị võ tướng của nhà Thục Hán vào cuối đời Đông Hán, đầu đời Tam Quốc trong lịch sử Trung Quốc. Ông là hậu duệ của Phục Ba tướng quân Mã Viện đời Đông Hán. Wikipedia

Sinh: 176 sau CN, Hưng Bình, Hàm Dương, Trung Quốc

Mất: 222 sau CN

Phụ huynh: Mã Đằng

Trẻ em: Ma Cheng, Ma Qiu

Anh chị em: Mã Thiết, Mã Hưu, Ma Yunlu

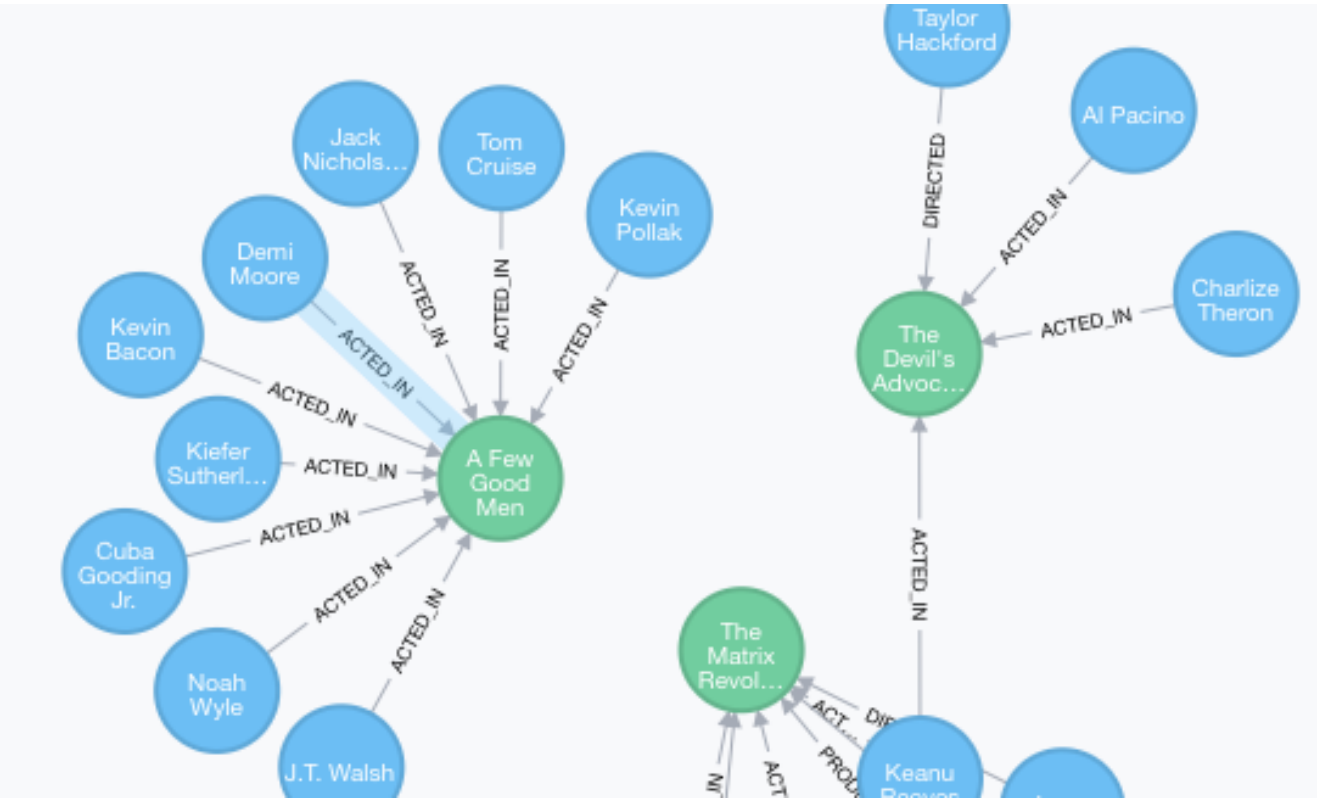
Mọi người cũng tìm kiếm

Xem hơn 15 mục khác

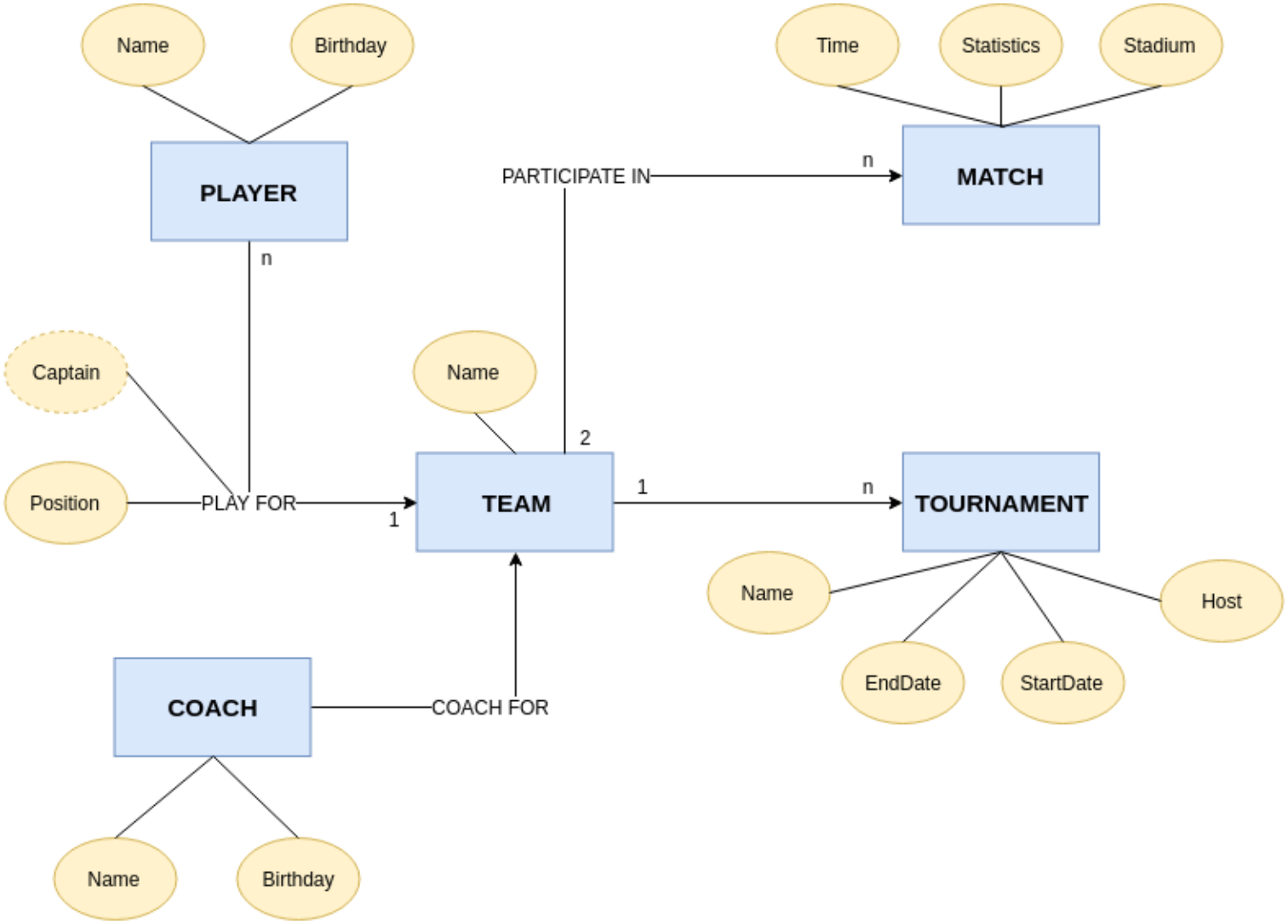
Triệu Vân | Hoàng Trung | Trương Phi | Lã Bố | Lưu Bị

Phản hồi

Knowledge graph in Neo4j



World Cup 2018 QA



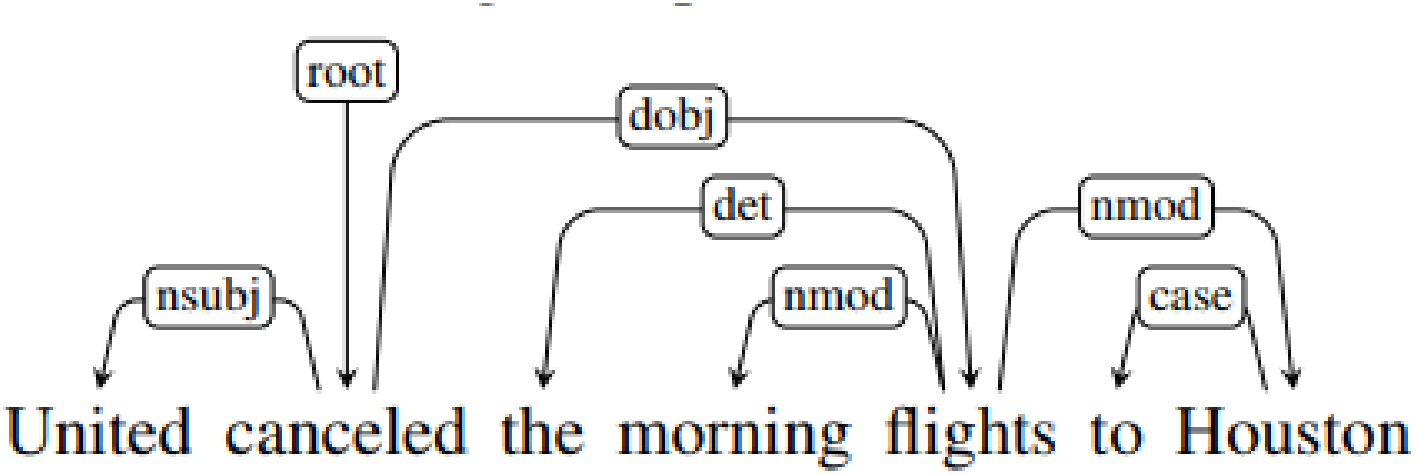
Our case

- World Cup QA is a closed domain fatoid QA
- The result need to be precised or at least we know what we don't know
- We don't have enough annotated data

Proposed method

- Analyze syntactic struture of the question on it's dependency tree
- Process on that tree to gather RDF Tripples
- Construct query from those tripples

Dependency parsing



Dependency parsing

Type of Dependencies

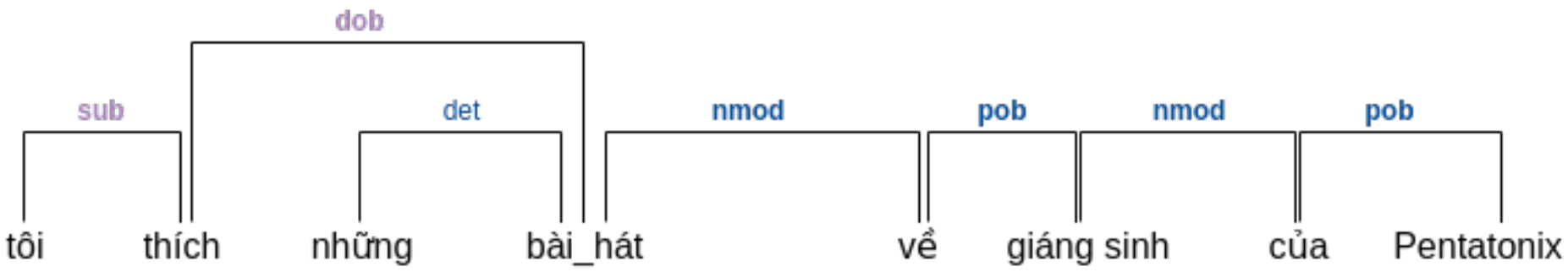
Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET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

Dependency parsing

Properties of dependency tree

- There is a single designed root node that has no incoming arcs.
- With the exception of the root node, each vertex has exactly one incoming arc.
- There is a unique path from the root node to each vertex in V.
- Dependency tree is projective.

Dependency parsing



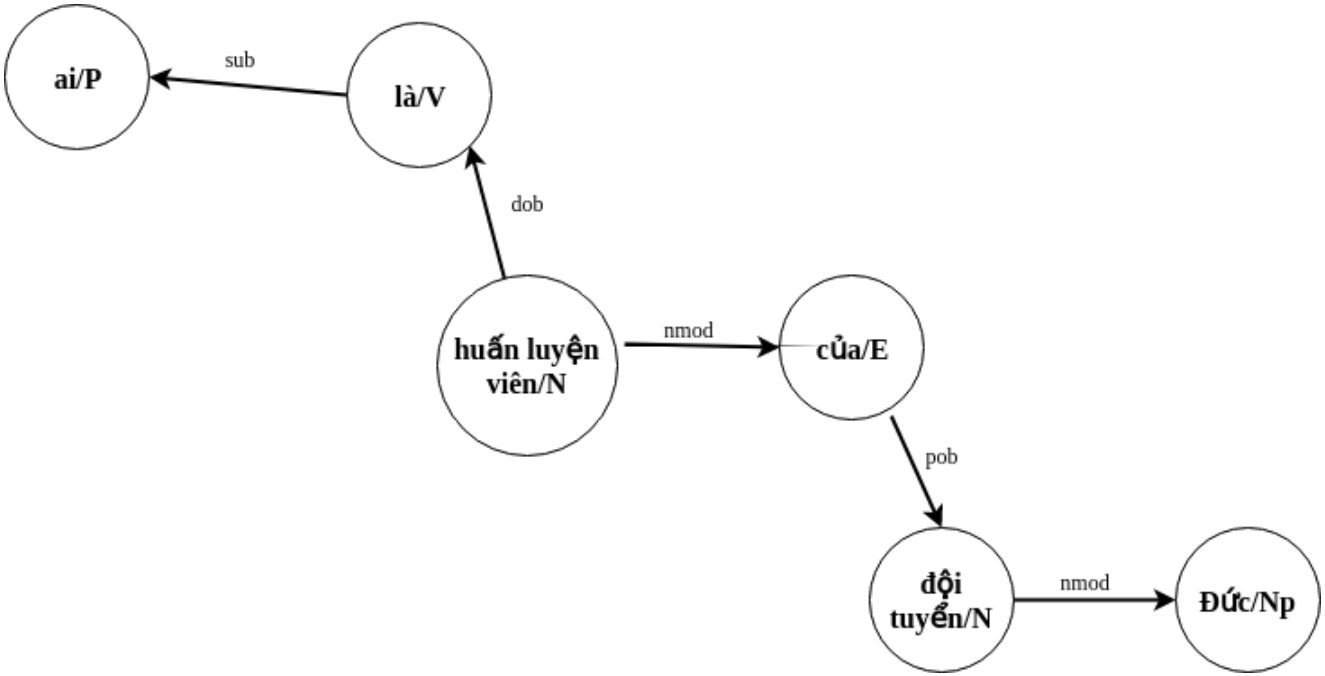
Pipeline - Step 1: Parsing

We feed the question through VnCoreNlpParser (https://github.com/vncorenlp/VnCoreNLP)

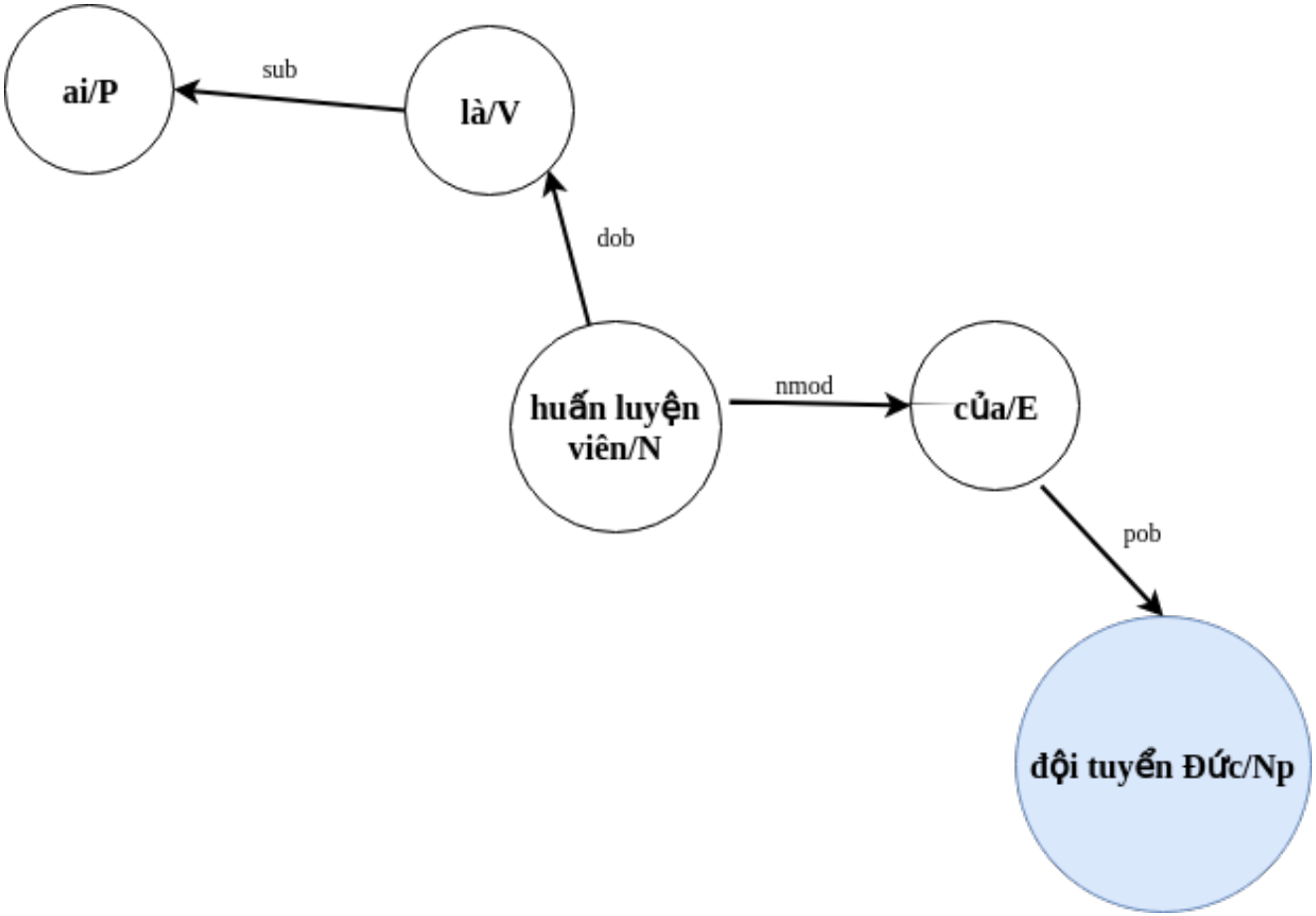
```
Ai là huấn luyện viên của đội tuyển Đức
1   Ai  P   O   2   sub
2   là  V   O   0   root
3   huấn_luyện_viên N   O   2   dob
4   của E   O   3   nmod
5   đội_tuyển   N   O   4   pob
6   Đức Np  B-PER   5   nmod
```

Pipeline - Step 1: Parsing

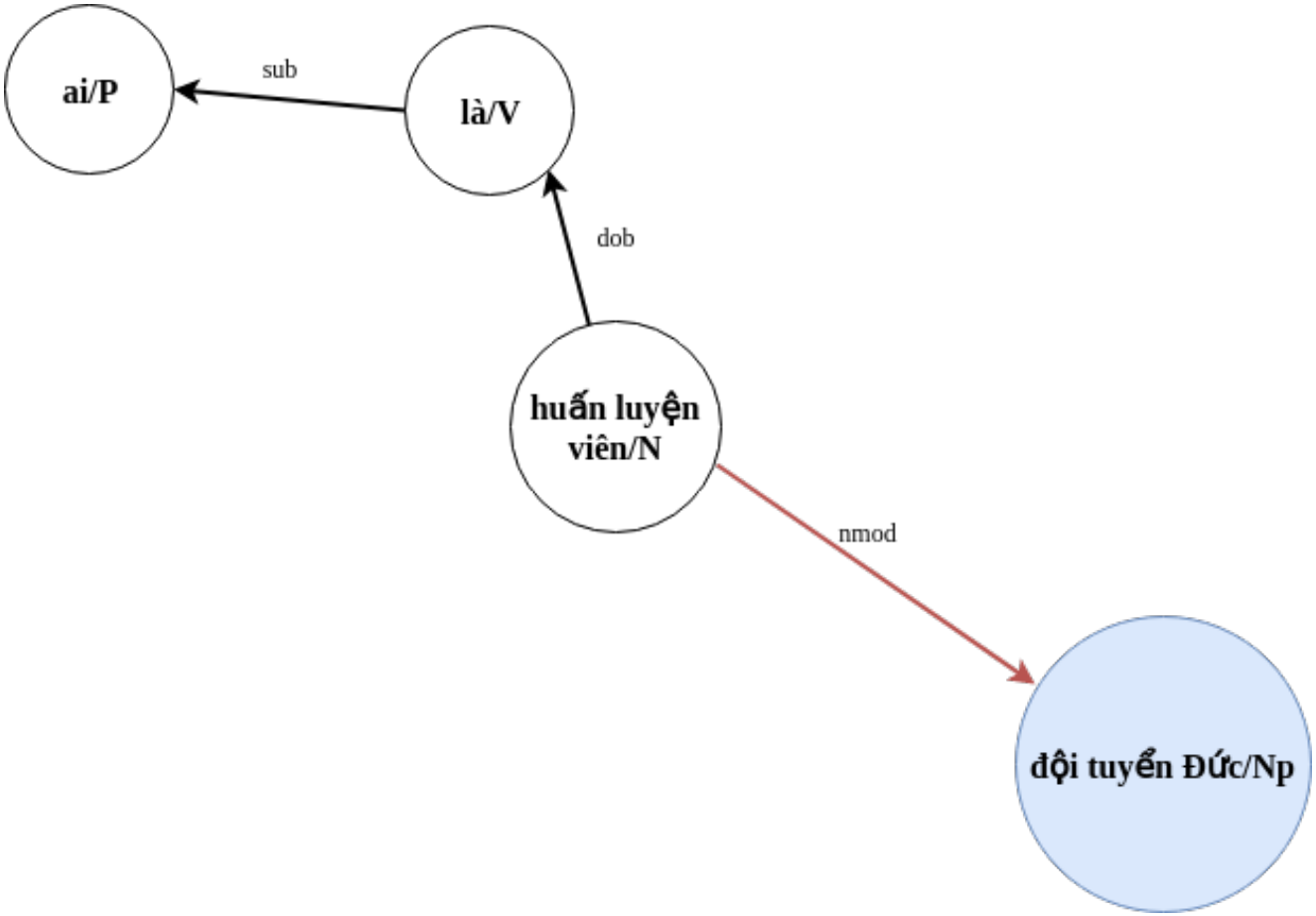
Q = "Huấn luyện viên đội tuyển Đức là ai ?"



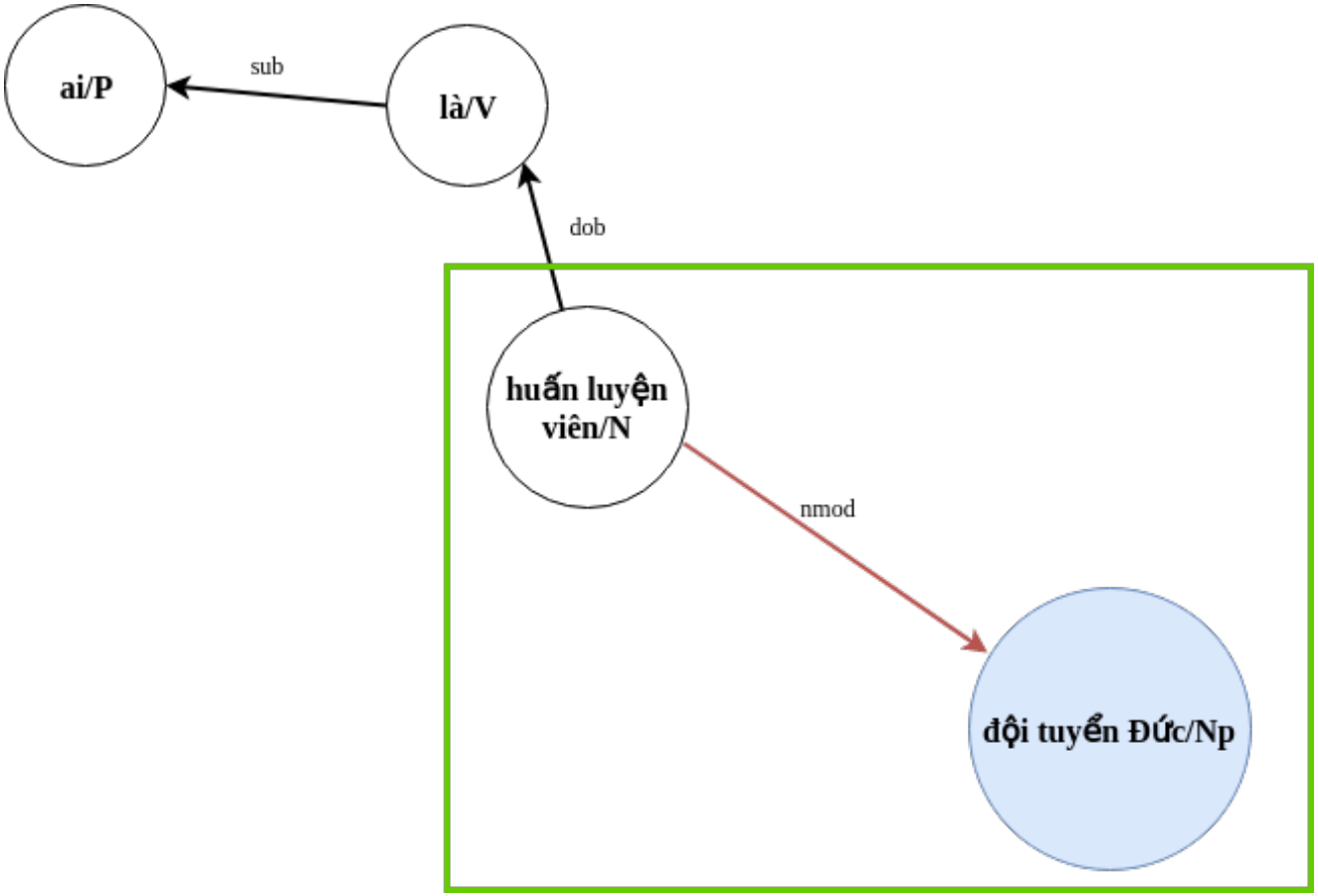
Pipeline - Step 2: Tree simplification



Pipeline - Step 2: Tree simplification



Pipeline - Step 2: Tree simplification

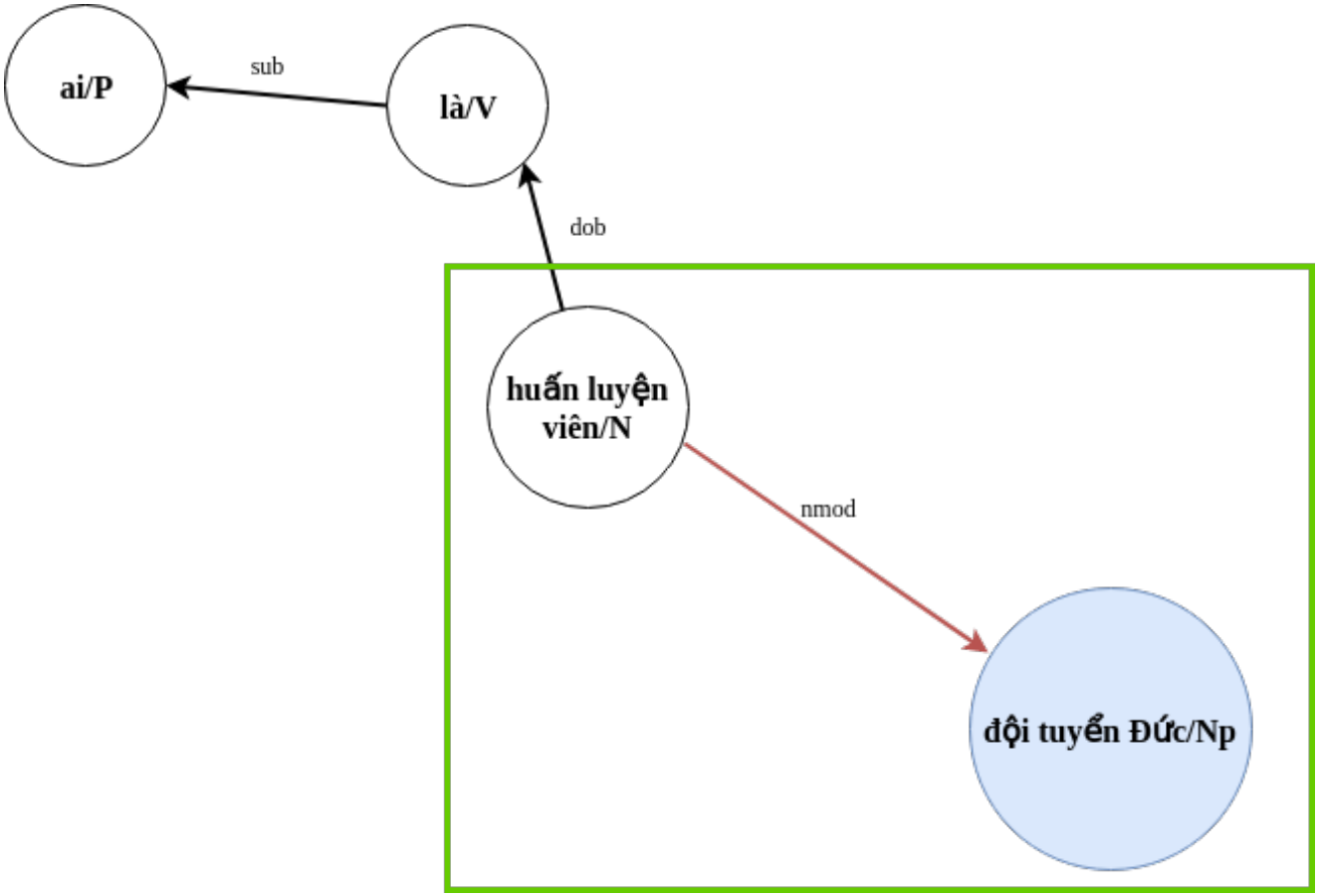


Pipeline - Step 3: RDF triples construction

- Assume the tree is in form of a linked list.
- Start with the leaf node whose tag is not **P** (question word)
- Iterate through nodes

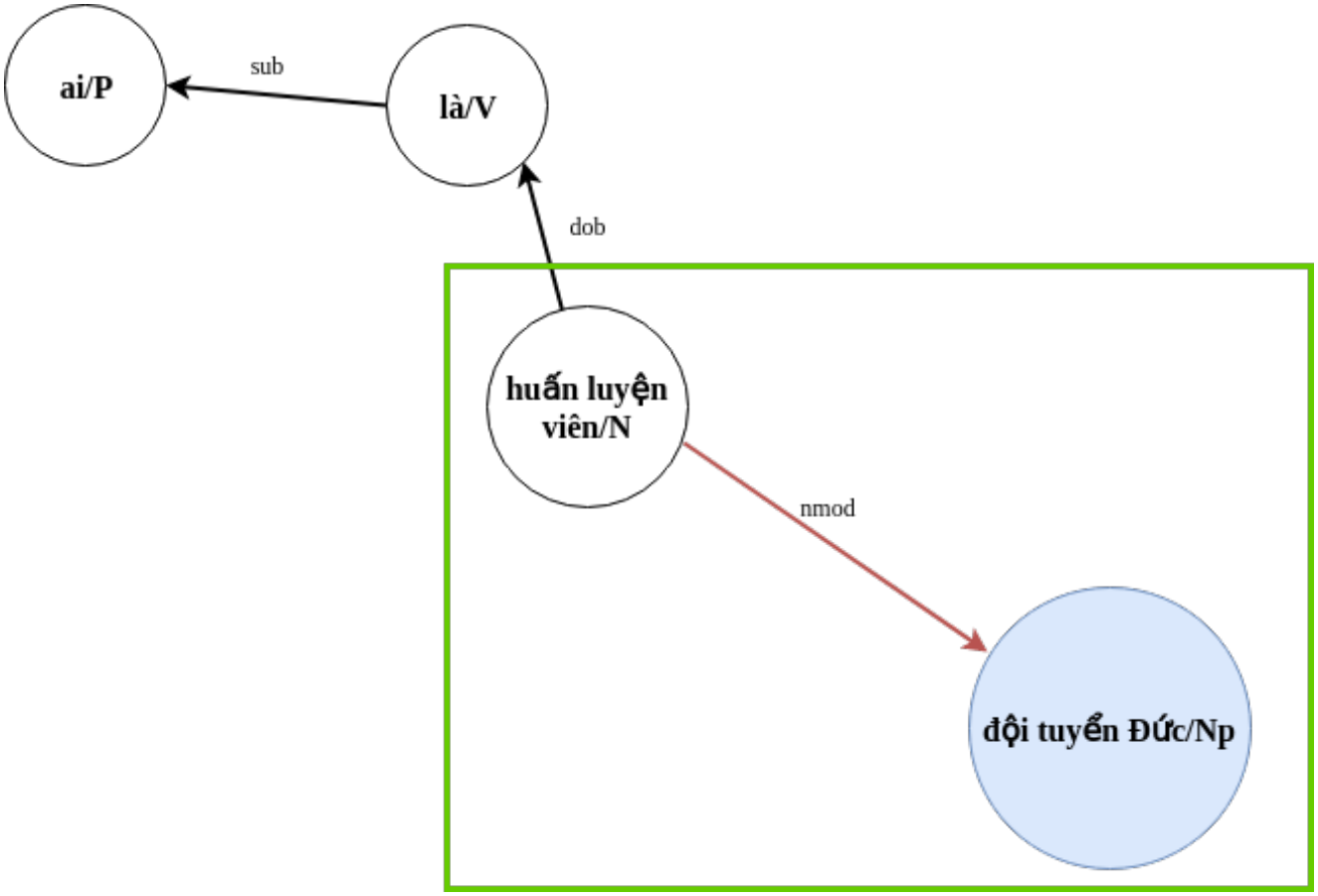
Pipeline - Step 3: RDF triples construction

T = (đội tuyển Đức,,)



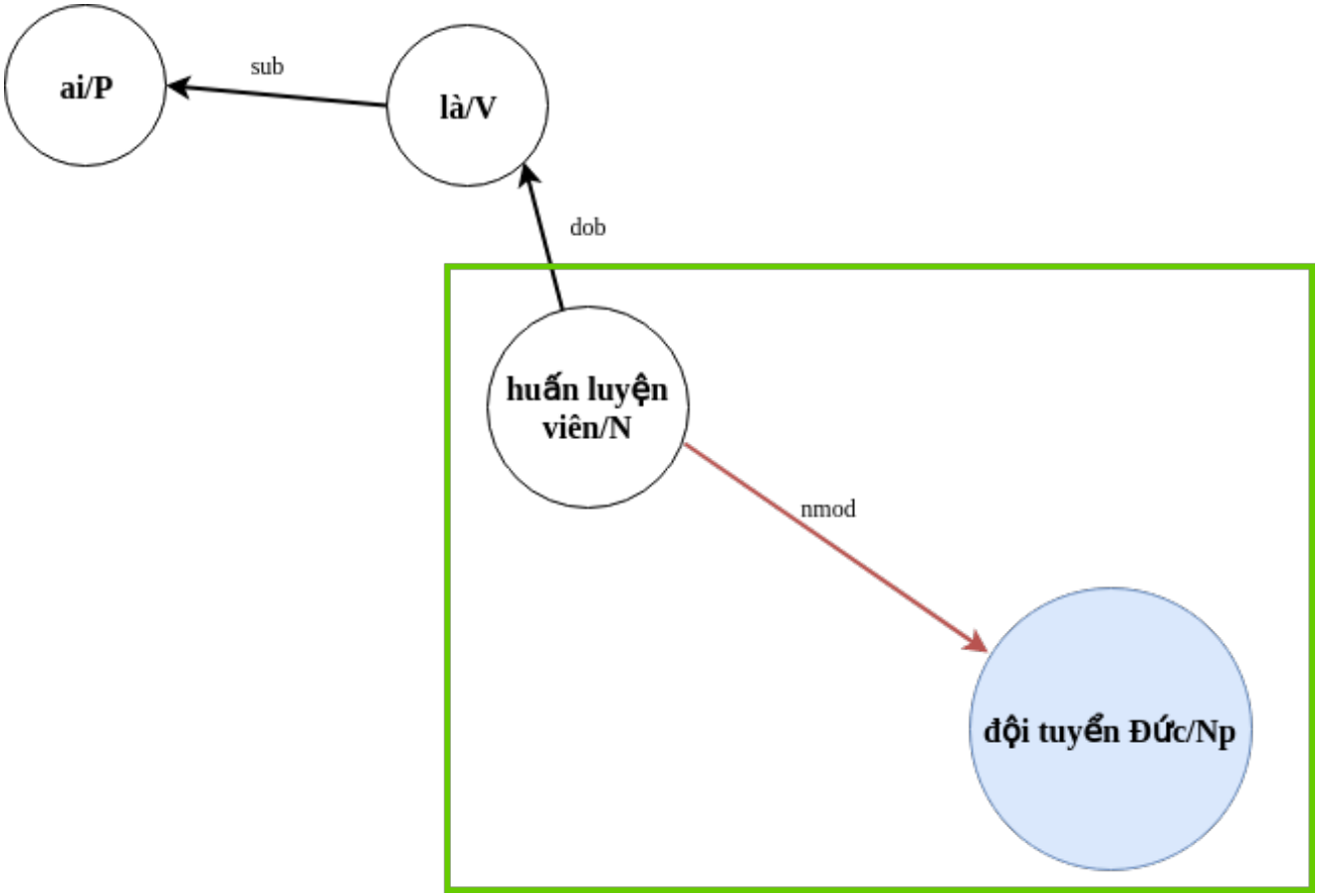
Pipeline - Step 3: RDF triples construction

T = (đội tuyển Đức, *VERBIFY(huấn luyện viên)*,)



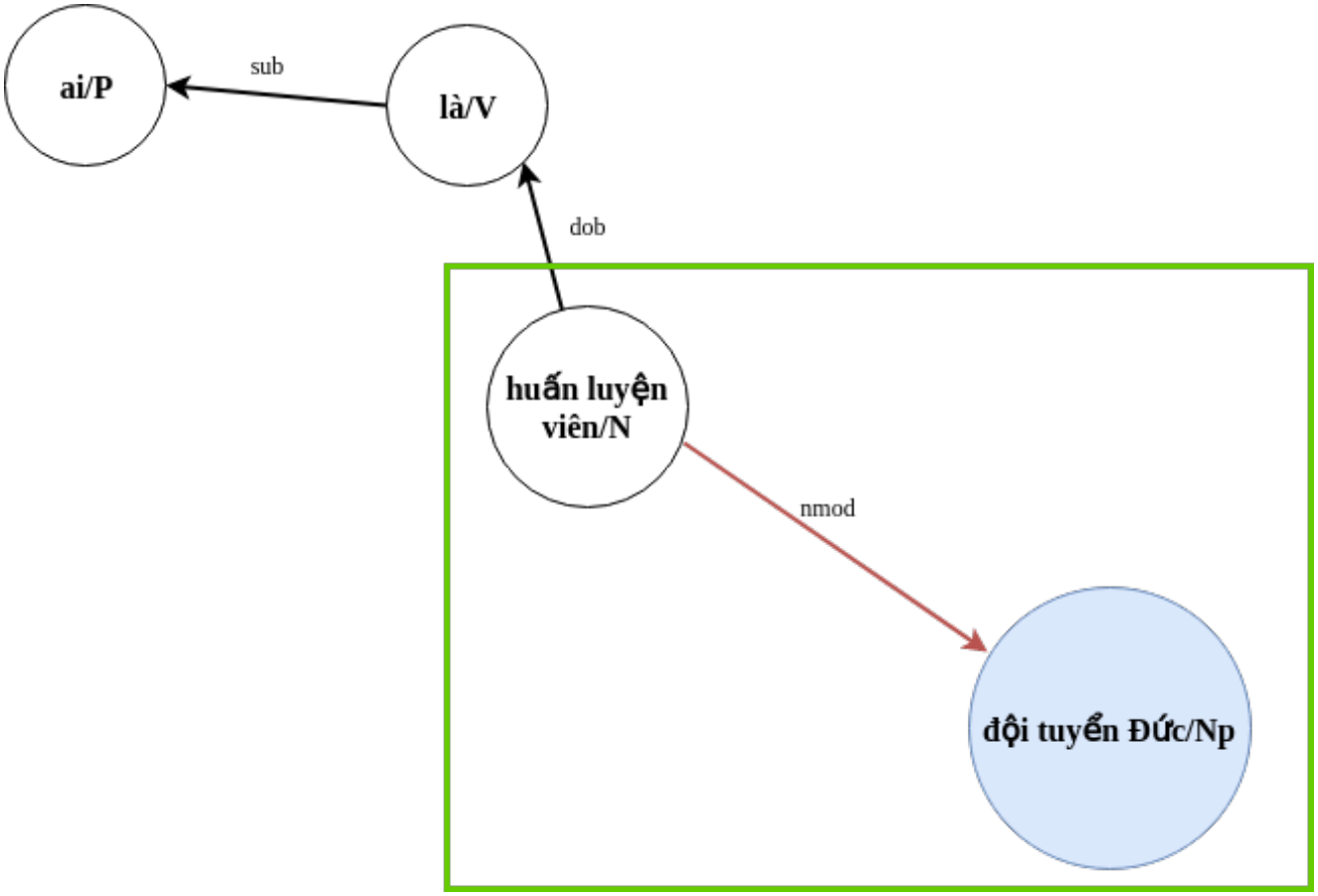
Pipeline - Step 3: RDF triples construction

T = (đội tuyển Đức, *COACH_FOR*,)



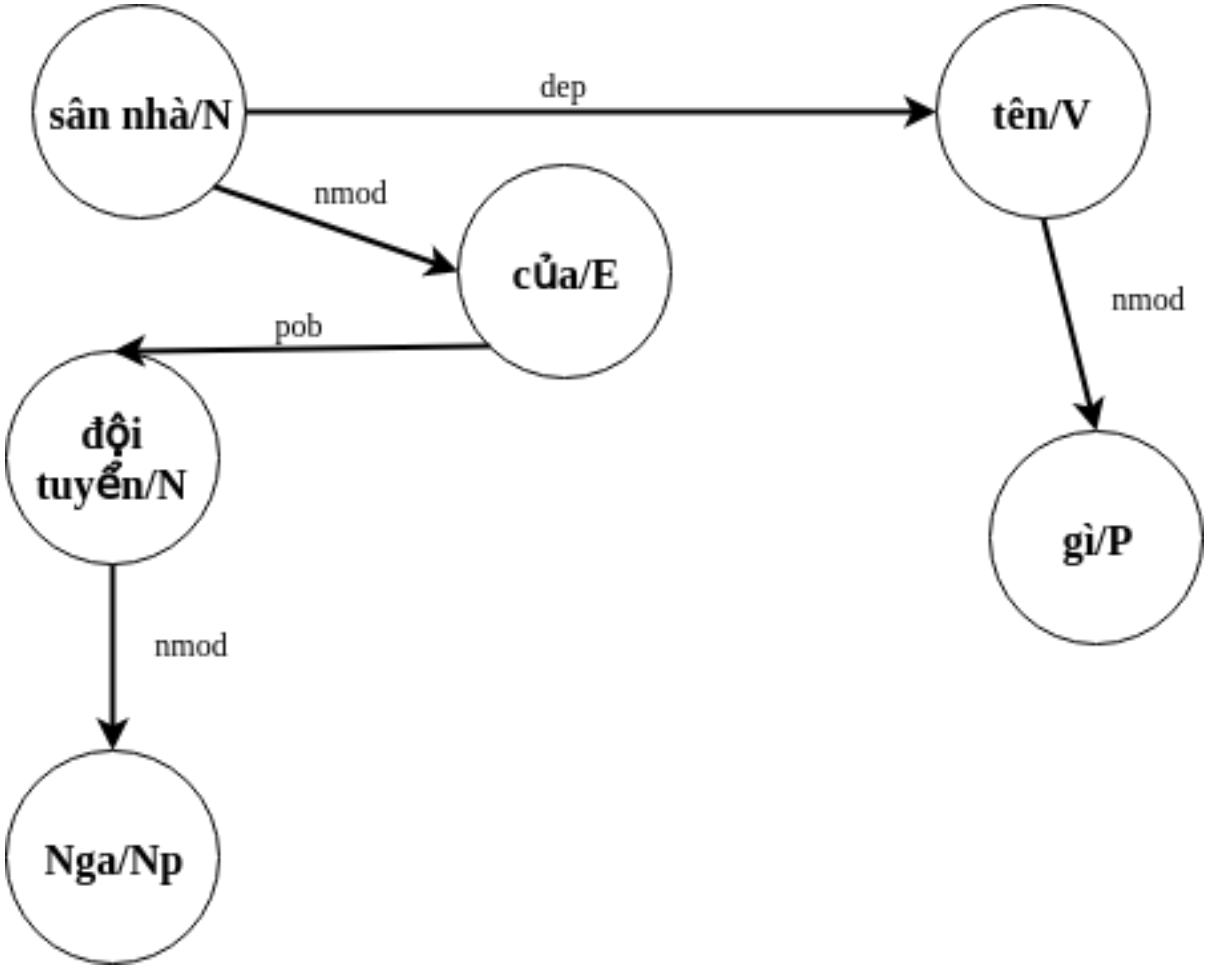
Pipeline - Step 3: RDF triples construction

T = (đội tuyển Đức, *COACH_FOR*, ?)



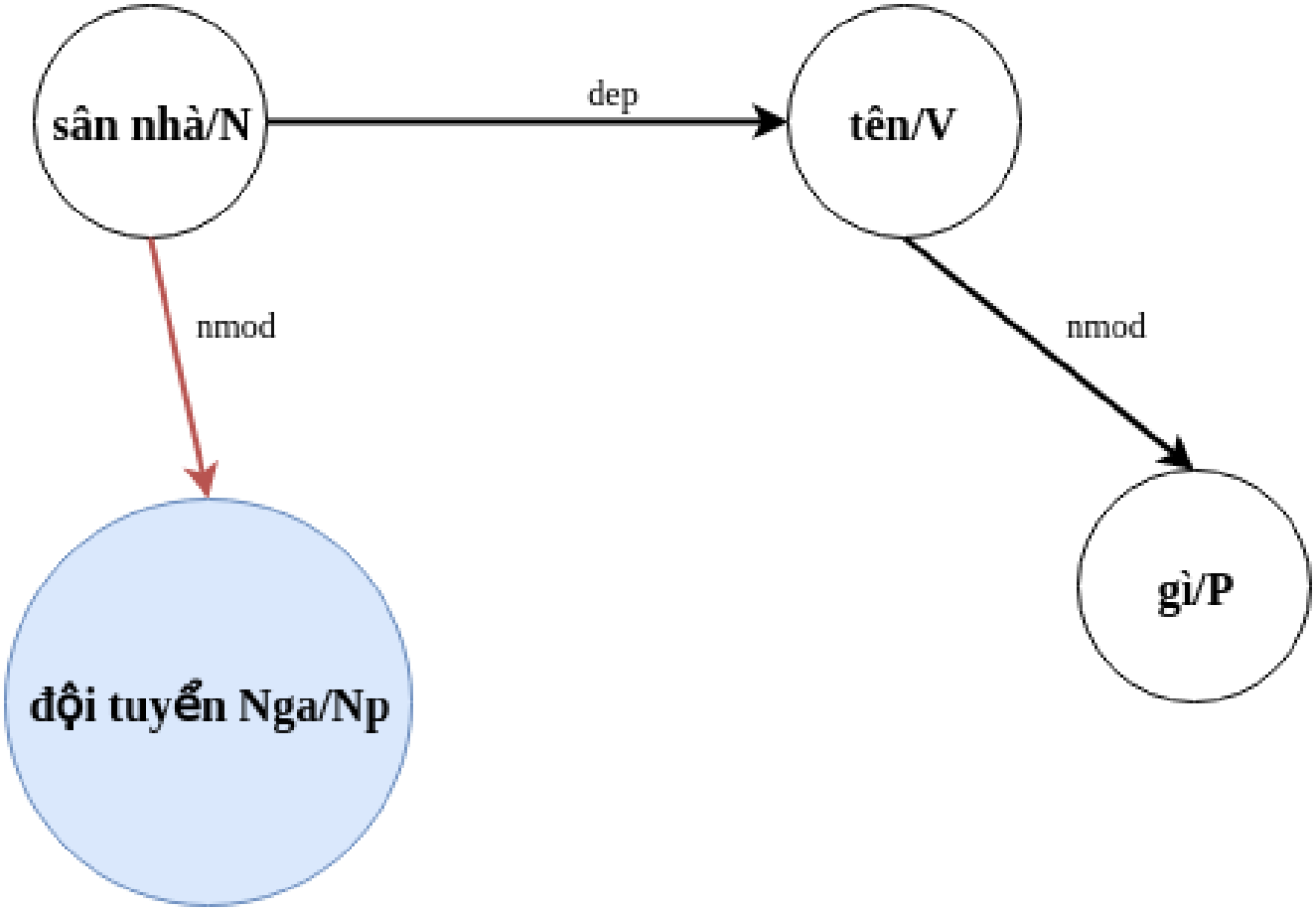
Examples

Q = "Sân nhà của đội tuyển Nga tên gì"



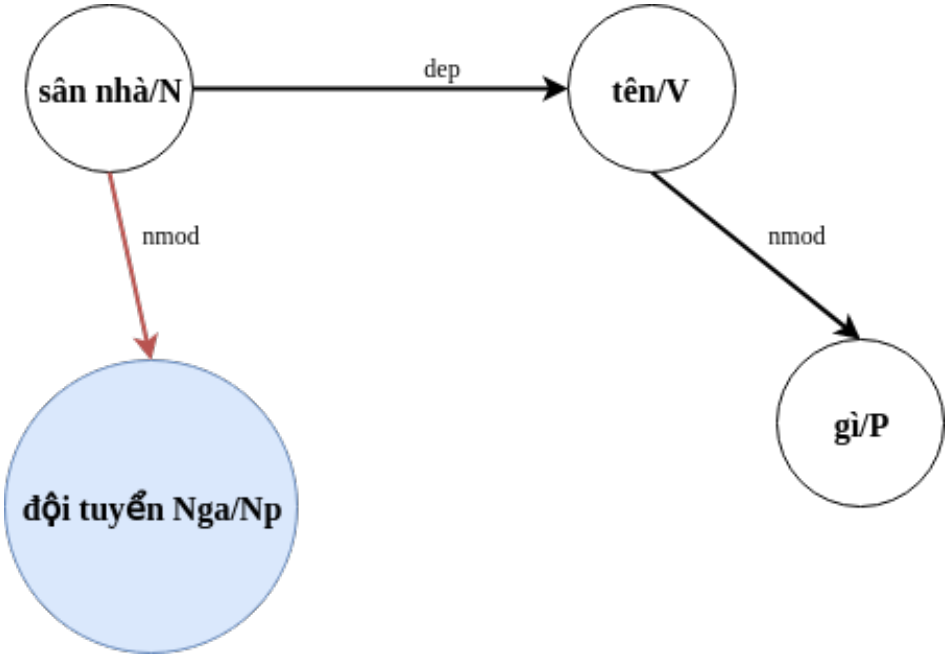
Examples

Q = "Sân nhà của đội tuyển Nga tên gì"



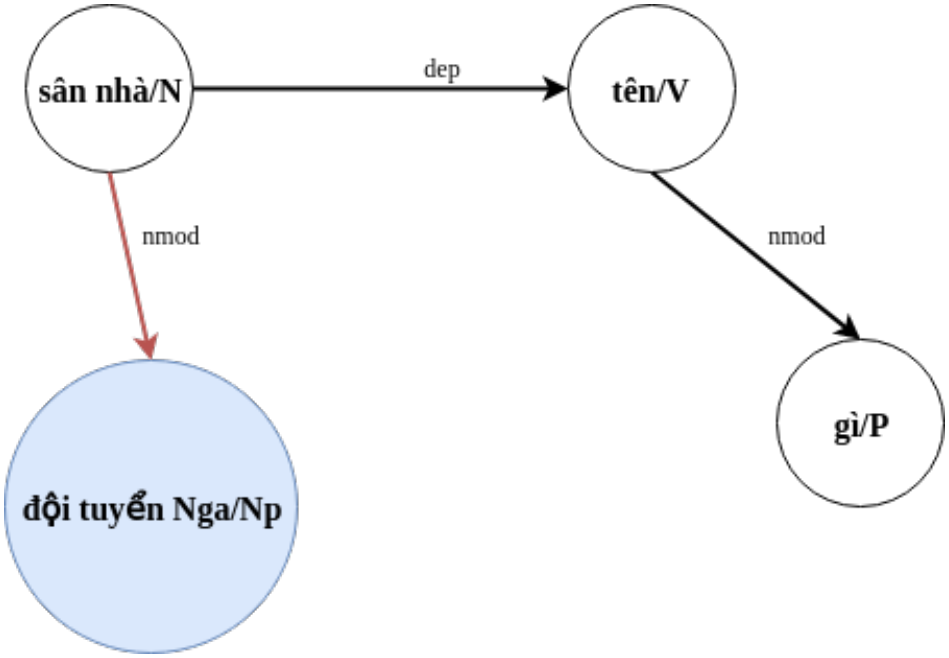
Examples

Q = "Sân nhà của đội tuyển Nga tên gì"
T_1 = (đội tuyển Nga, VERBIFY(sân nhà),)



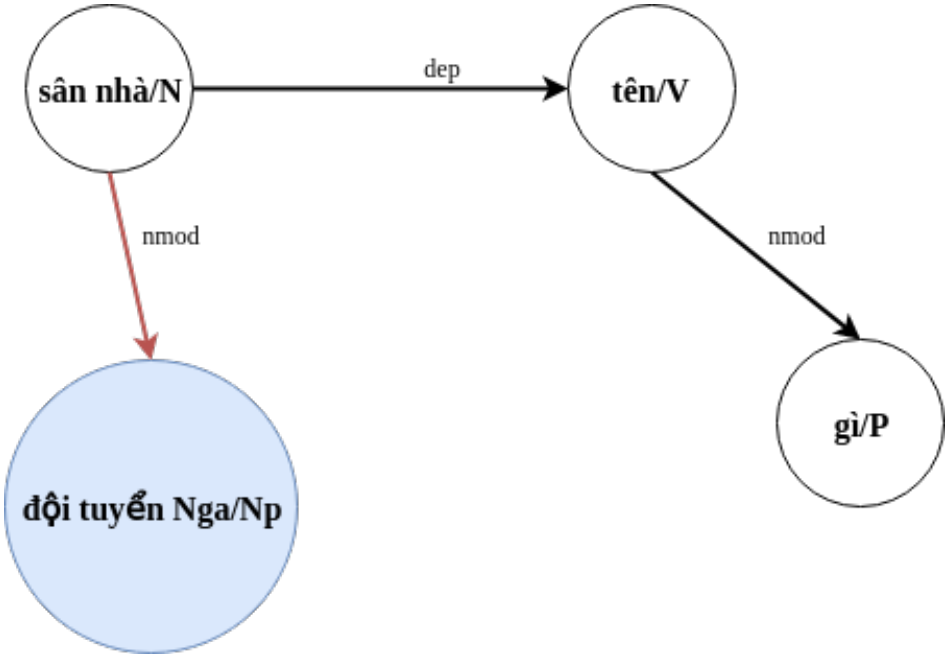
Examples

Q = "Sân nhà của đội tuyển Nga tên gì"
T_1 = (đội tuyển Nga, IS_HOME_STADIUM, ?)



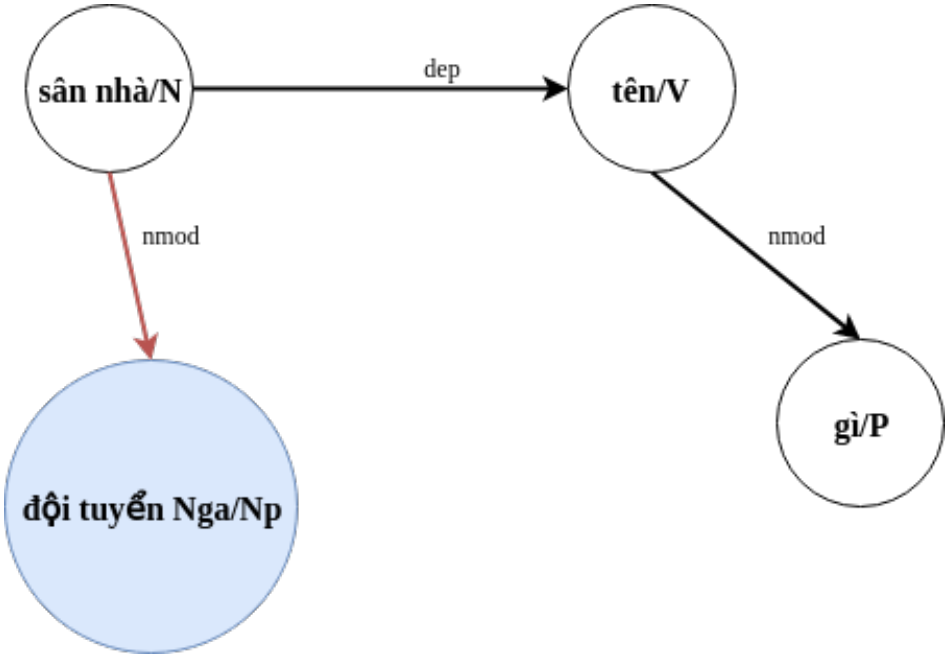
Examples

Q = "Sân nhà của đội tuyển Nga tên gì"
T_1 = (đội tuyển Nga, IS_HOME_STADIUM, ?)
T_2 = (<*prev*>,,)



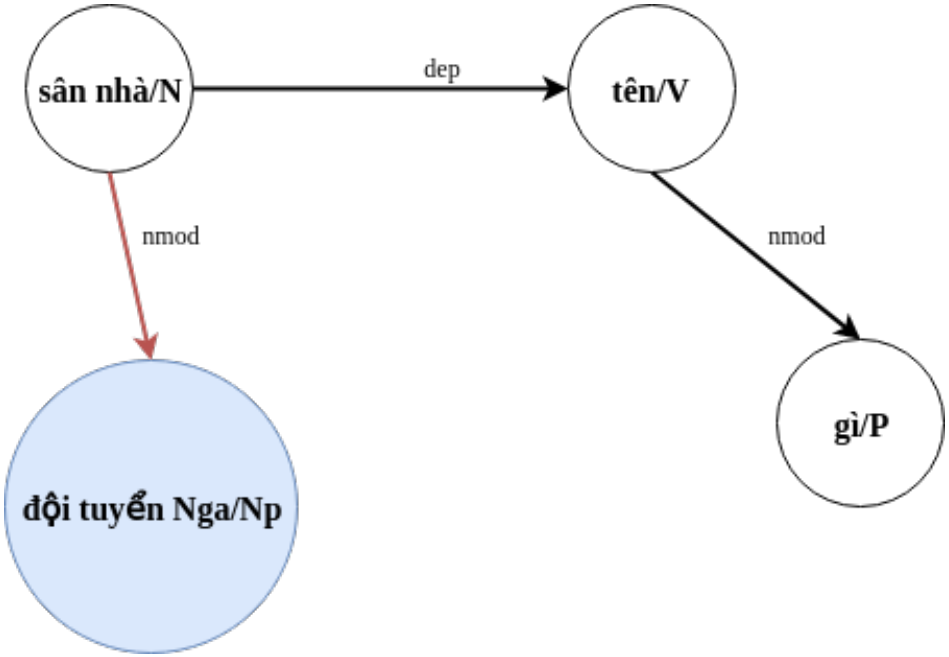
Examples

Q = "Sân nhà của đội tuyển Nga tên gì"
T_1 = (đội tuyển Nga, IS_HOME_STADIUM, ?)
T_2 = (<*prev*>,,)
NOUNLIFY(tên/V) = tên/N



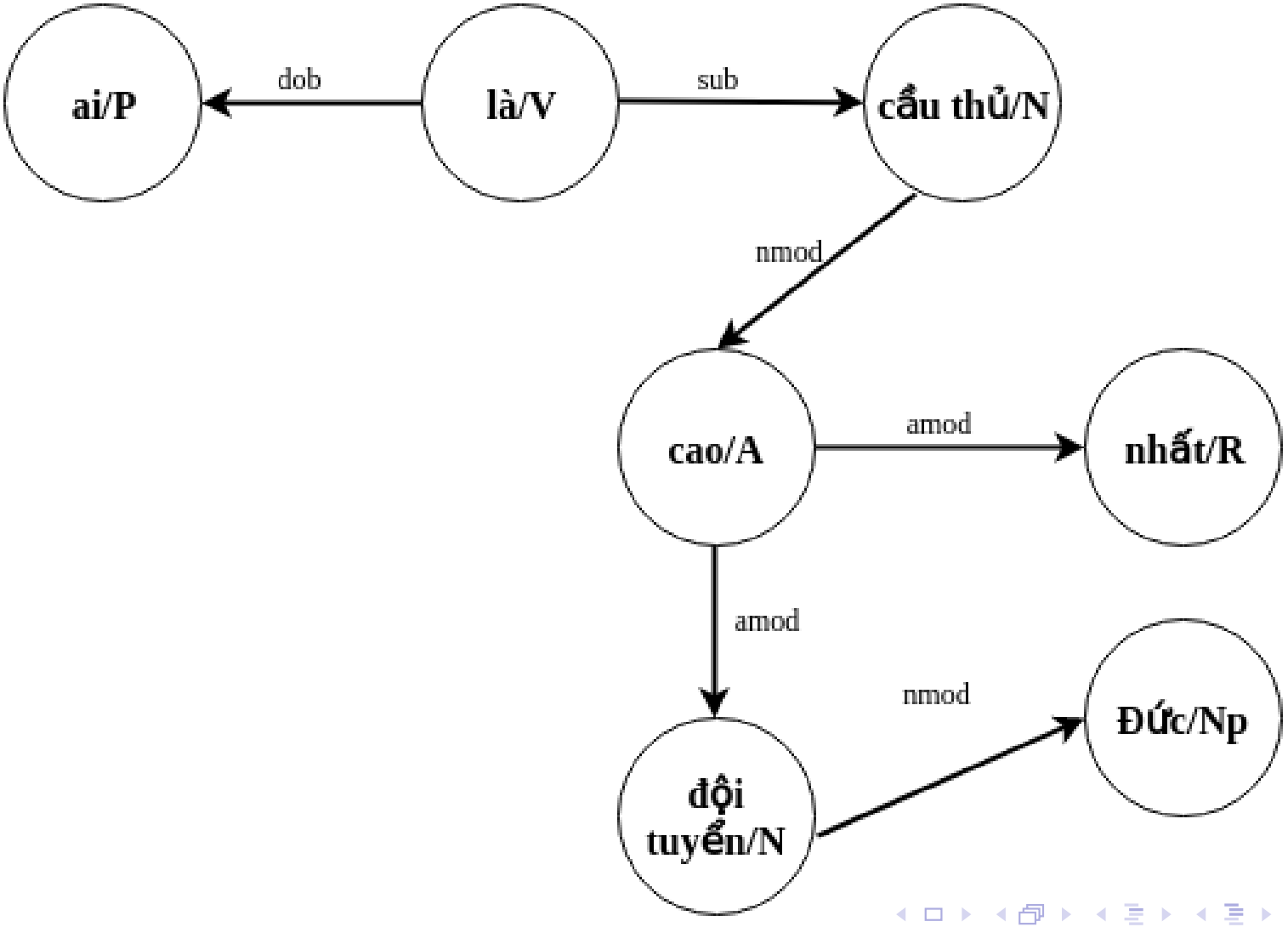
Examples

Q = "Sân nhà của đội tuyển Nga tên gì"
T_1 = (đội tuyển Nga, IS_HOME_STADIUM, ?)
T_2 = (<*prev*>.tên,,)



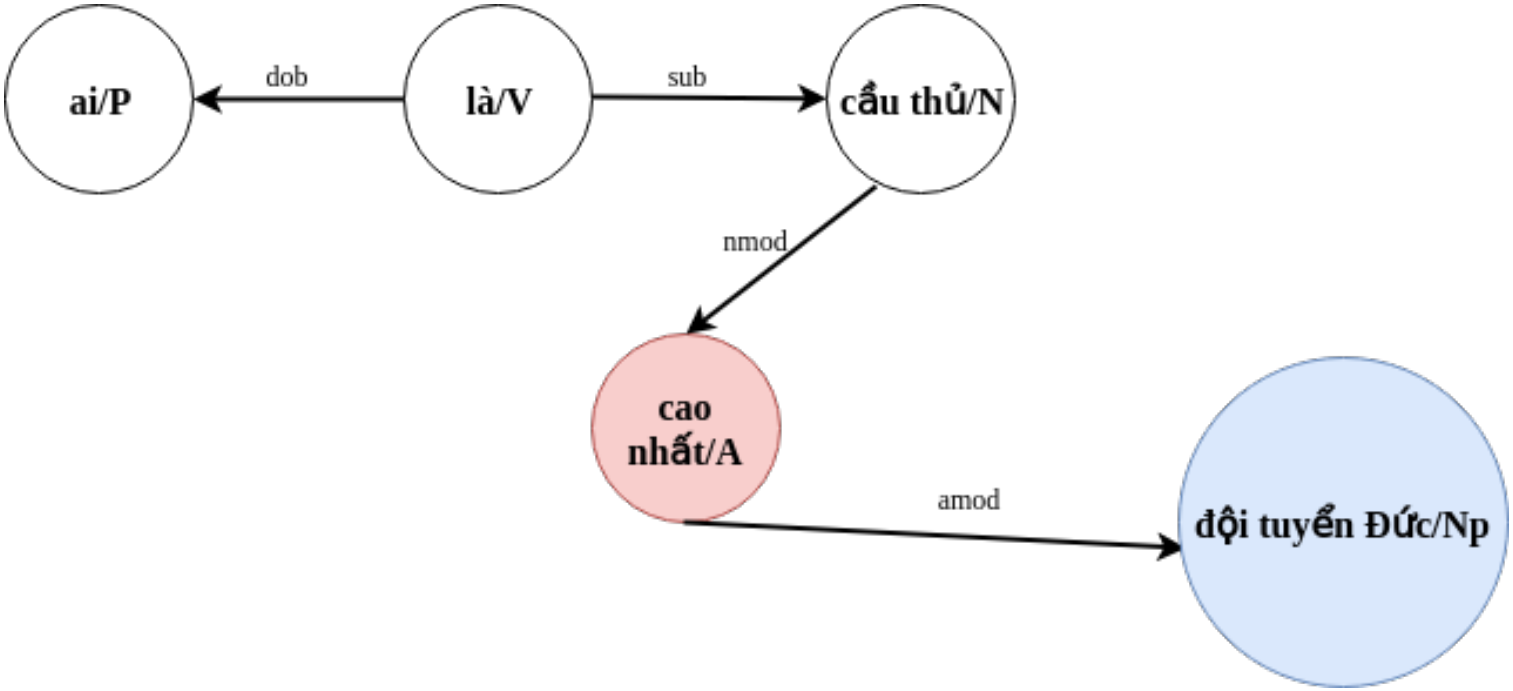
Examples

Q = "Cầu thủ cao nhất đội tuyển Đức là ai"



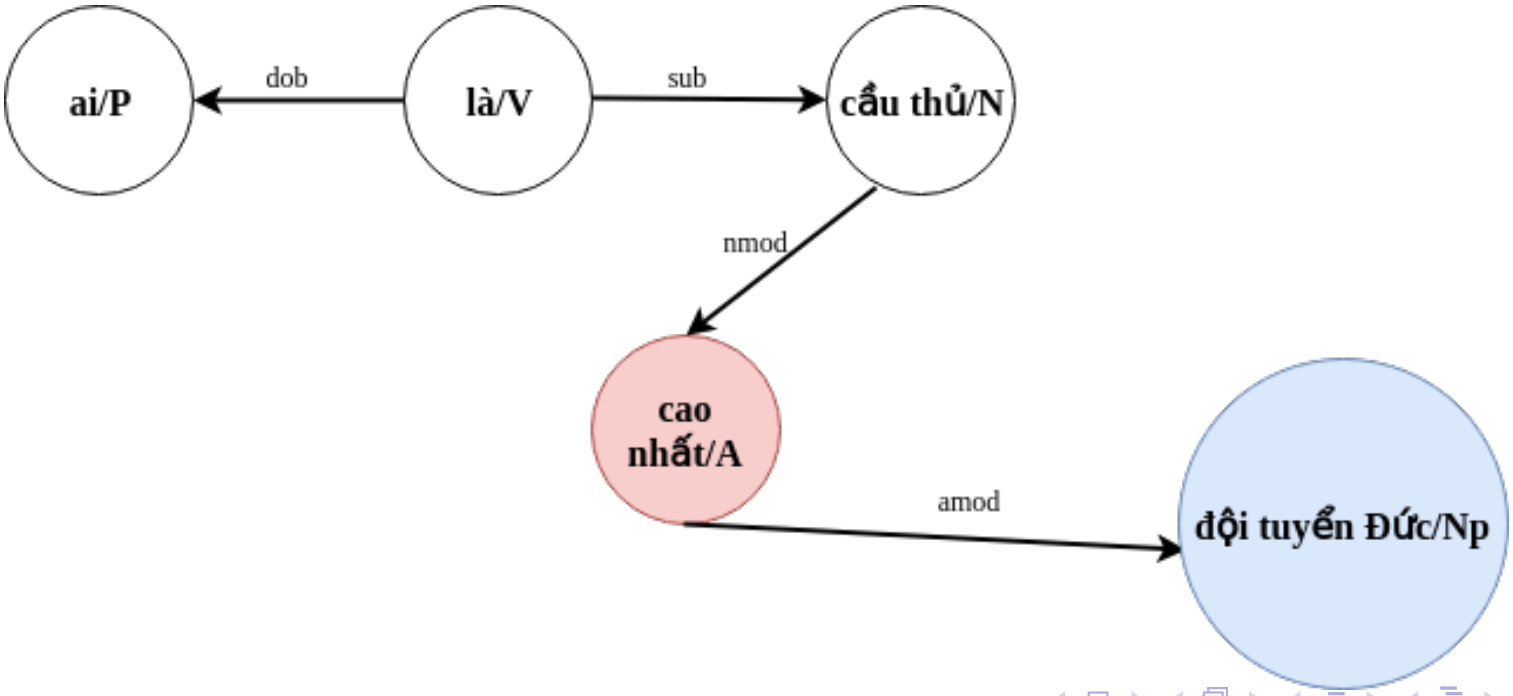
Examples

Q = "Cầu thủ cao nhất đội tuyển Đức là ai"
T = (đội tuyển Đức,,)
We meet the adjective "cao nhất", we will NOUNLIFY the adjective "cao", if we success, then append a slot to T.
T = (đội tuyển Đức,,,NOUNLIFY(cao))



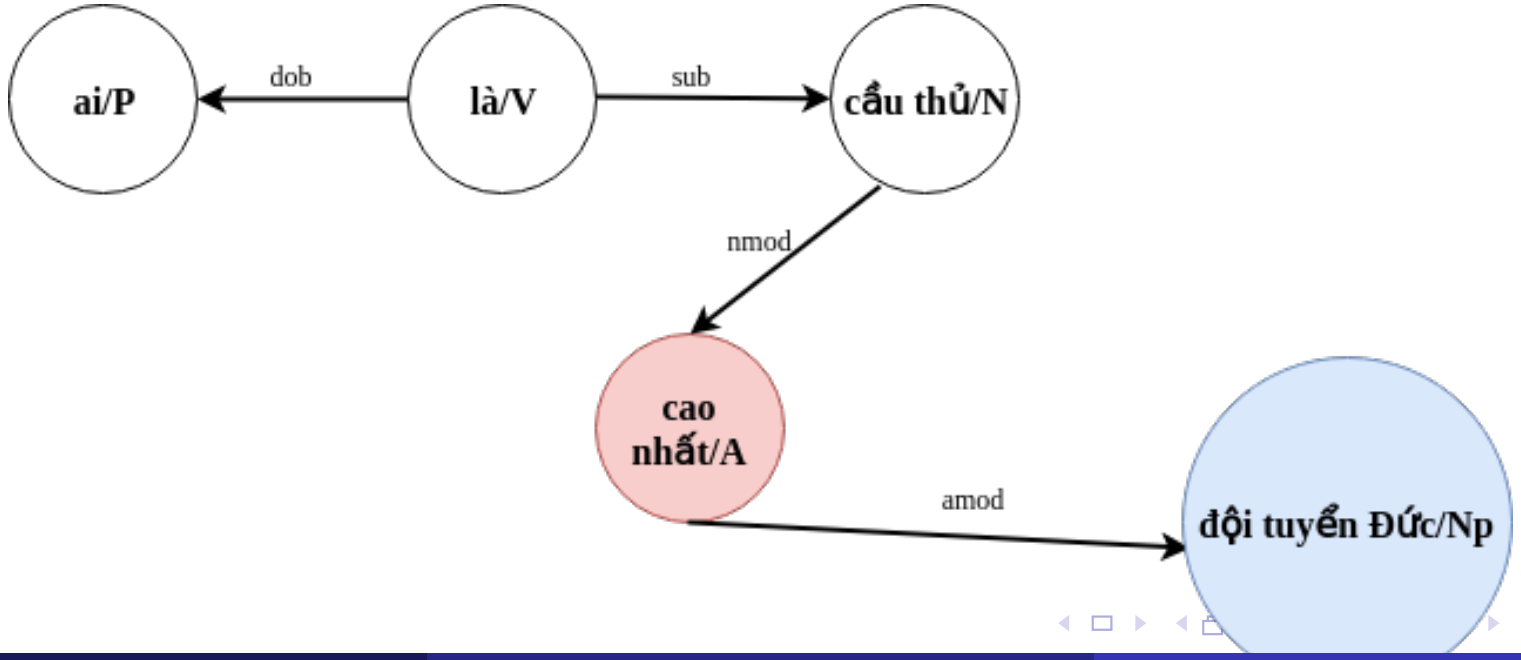
Examples

Q = "Cầu thủ cao nhất đội tuyển Đức là ai"
T = (đội tuyển Đức,,)
We meet the adjective "cao nhất", we will NOUNLIFY the adjective "cao", if we success, then append a slot to T.
T = (đội tuyển Đức,,,NOUNLIFY(cao))
T = (đội tuyển Đức,,,chiều cao)



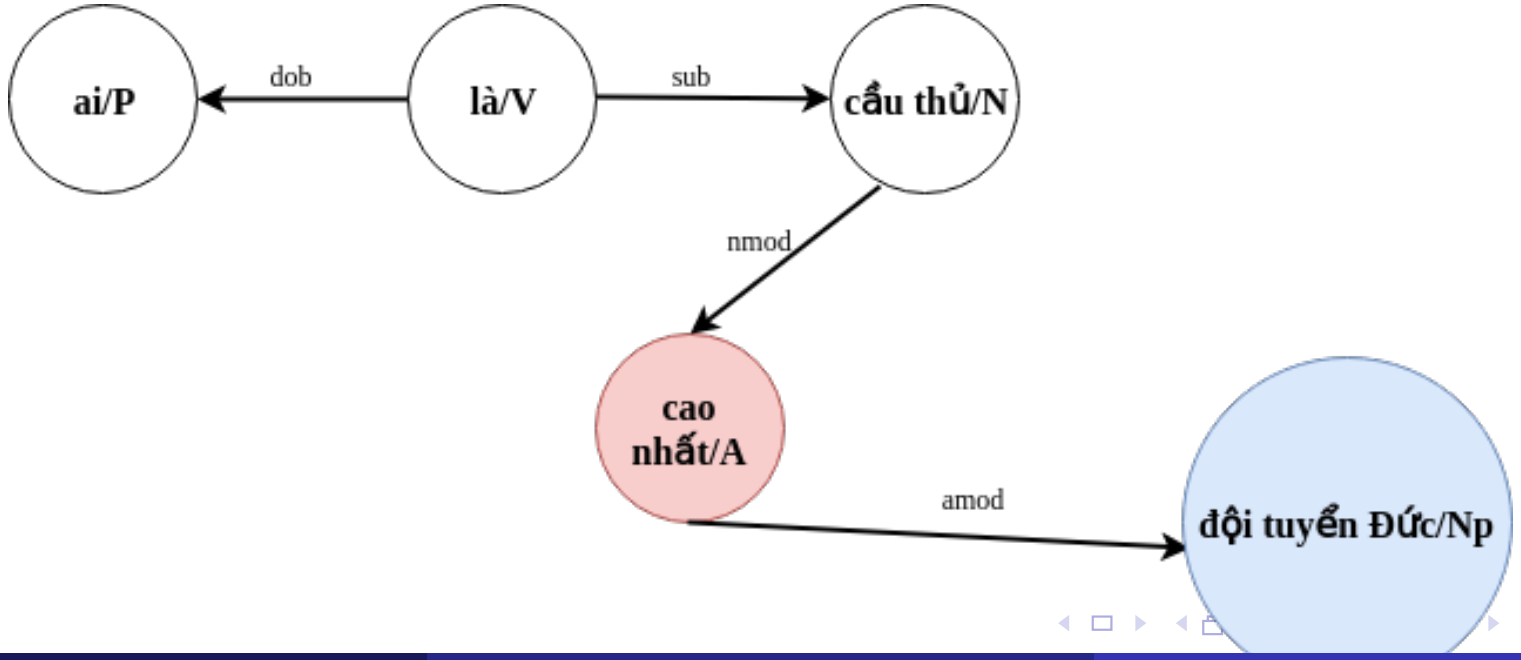
Examples

Q = "Cầu thủ cao nhất đội tuyển Đức là ai"
T = (đội tuyển Đức,,)
We meet the adjective "cao nhất", we will NOUNLIFY the adjective "cao", if we success, then append a slot to T.
T = (đội tuyển Đức,,,NOUNLIFY(cao))
T = (đội tuyển Đức,,,chiều cao)
T = (đội tuyển Đức,VERBIFY(cầu thủ),,chiều cao)



Examples

Q = "Cầu thủ cao nhất đội tuyển Đức là ai"
T = (đội tuyển Đức,,)
We meet the adjective "cao nhất", we will NOUNLIFY the adjective "cao", if we success, then append a slot to T.
T = (đội tuyển Đức,,,NOUNLIFY(cao))
T = (đội tuyển Đức,,,chiều cao)
T = (đội tuyển Đức,PLAY_FOR, ? ,chiều cao)



About this method

- The quality of the method depends on:
  - How good the parser is
  - How well our dictionary (VERBIFY, NOUNLIFY) is defined
- Advantages:
  - Require no human annotation
  - Data driven
  - Algorithmic & Debuggable
  - Stable & extensible
  - We know what we know and what we don't know
- Disavantages:
  - Require understanding about language (Vietnamese)
  - Require understanding about the domain (Football)
  - Rely on 3rd parties (VnCoreNlp)
  - When the parser goes wrong, we have to hard-code

Tham khảo

- Yassine Hamoudi & Tom Cornebize: Extracting RDF triples using the Stanford Parser
- Daniel Jurafsky & James H. Martin: Speech and Language Processing
- Dennis Diefenbach, Vanessa Lopez, Kamal Singh & Pierre Maret: Core Techniques of Question Answering Systems over Knowledge Bases: a Survey Dennis