

✓ Phần 1: Giới thiệu và Cài đặt

```
pip install -U spacy
```

```
Requirement already satisfied: spacy in c:\users\adm\anaconda3\lib\site-packages (3.8.11)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (1.0.15)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (2.0.13)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (3.0.12)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (8.3.10)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (1.1.3)
Requirement already satisfied: srslly<3.0.0,>=2.4.3 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (2.5.2)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (0.4.3)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (0.20.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (4.67.1)
Requirement already satisfied: numpy>=1.19.0 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (2.3.5)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (2.32.5)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in c:\users\adm\anaconda3\lib\site-packages (from spacy)
Requirement already satisfied: jinja2 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (3.1.6)
Requirement already satisfied: setuptools in c:\users\adm\anaconda3\lib\site-packages (from spacy) (80.9.0)
Requirement already satisfied: packaging>=20.0 in c:\users\adm\anaconda3\lib\site-packages (from spacy) (25.0)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\adm\anaconda3\lib\site-packages (from pydantic!=1.8,!=1.8.1)
Requirement already satisfied: pydantic-core==2.20.1 in c:\users\adm\anaconda3\lib\site-packages (from pydantic!=1.8,!=1.8.1)
Requirement already satisfied: typing-extensions>=4.6.1 in c:\users\adm\anaconda3\lib\site-packages (from pydantic!=1.8,!=1.8.1)
Requirement already satisfied: charset_normalizer<4,>=2 in c:\users\adm\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0)
Requirement already satisfied: idna<4,>=2.5 in c:\users\adm\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already satisfied: urlib3<3,>=1.21.1 in c:\users\adm\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\adm\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already satisfied: blis<1.4.0,>=1.3.0 in c:\users\adm\anaconda3\lib\site-packages (from thinc<8.4.0,>=8.3.4->spacy)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in c:\users\adm\anaconda3\lib\site-packages (from thinc<8.4.0,>=8.3.4->spacy)
Requirement already satisfied: colorama in c:\users\adm\anaconda3\lib\site-packages (from tqdm<5.0.0,>=4.38.0->spacy) (0.4.6)
Requirement already satisfied: click>=8.0.0 in c:\users\adm\anaconda3\lib\site-packages (from typer-slim<1.0.0,>=0.3.0->spacy)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in c:\users\adm\anaconda3\lib\site-packages (from weasel<0.5.0,>=0.4.2)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in c:\users\adm\anaconda3\lib\site-packages (from weasel<0.5.0,>=0.4.2)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\adm\anaconda3\lib\site-packages (from jinja2->spacy) (3.0.3)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
```

✓ Phần 2: Phân tích câu và Trực quan hóa

✓ 2.1. Tải mô hình và phân tích câu

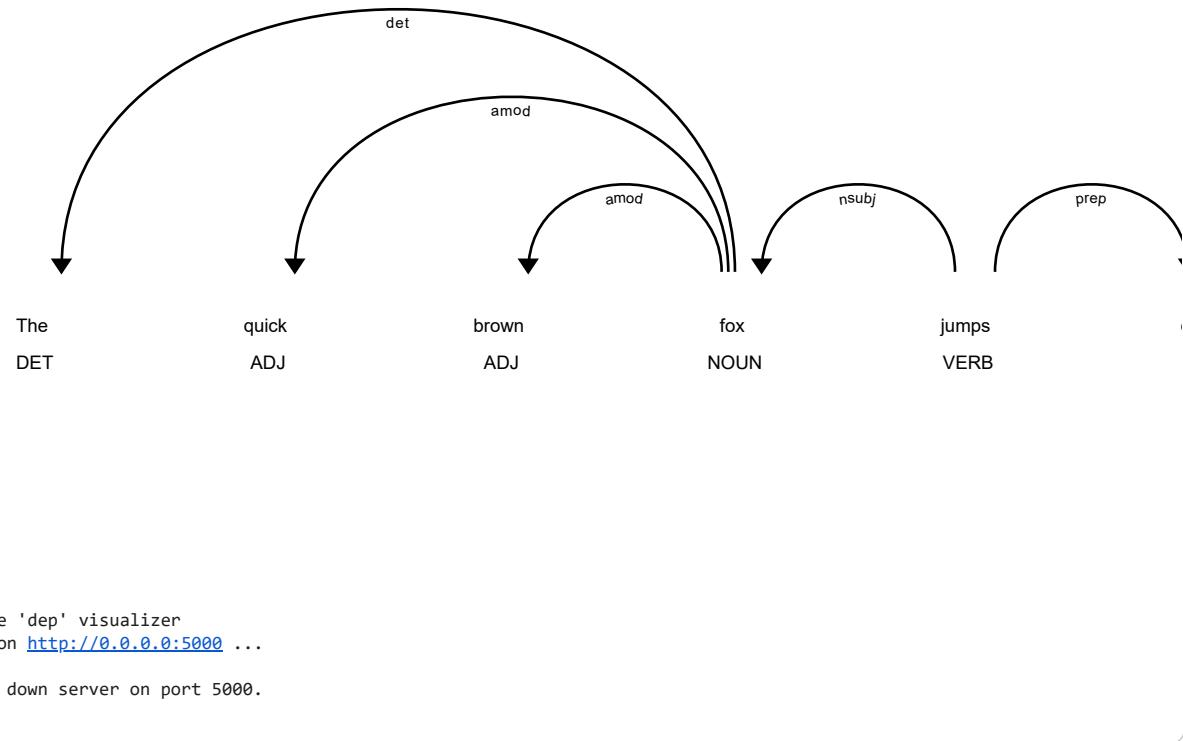
```
import spacy
from spacy import displacy

# Tải mô hình tiếng Anh đã cài đặt
# Sử dụng en_core_web_md vì nó chứa các vector từ và cây cú pháp đầy đủ
nlp = spacy.load("en_core_web_md")
# Câu ví dụ
text = "The quick brown fox jumps over the lazy dog."
# Phân tích câu với pipeline của spaCy
doc = nlp(text)
```

✓ 2.2. Trực quan hóa cây phụ thuộc

```
# Tùy chọn để hiển thị trong trình duyệt
options = {"compact": True, "color": "blue", "font": "Source Sans Pro"}
# Khởi chạy server tại http://127.0.0.1:5000
# Bạn có thể truy cập địa chỉ này trên trình duyệt để xem cây phụ thuộc
# Nhấn Ctrl+C trong terminal để dừng server
displacy.serve(doc, style="dep")
```

```
c:\Users\ADM\anaconda3\Lib\site-packages\spacy\displacy\__init__.py:108: UserWarning: [W011] It looks like you're calling di
warnings.warn(Warnings.W011)
```



Trả lời câu hỏi:

1. Từ nào là gốc (ROOT) của câu?

Từ **jumps** là gốc (ROOT) của câu "The quick brown fox jumps over the lazy dog."

2. **jumps có những từ phụ thuộc (dependent) nào? Các quan hệ đó là gì?**

Từ phụ thuộc	Quan hệ cú pháp	Mô tả
fox	nsubj (nominal subject)	Chủ ngữ danh từ của động từ
over	prep (prepositional modifier)	Cụm giới từ bổ nghĩa cho động từ

3. **fox là head của những từ nào?**

Từ phụ thuộc	Quan hệ cú pháp	Mô tả
The	det (determiner)	Mạo từ xác định
quick	amod (adjectival modifier)	Tính từ bổ nghĩa danh từ
brown	amod (adjectival modifier)	Tính từ bổ nghĩa danh từ

▼ Phần 3: Truy cập các thành phần trong cây phụ thuộc

```
# Lấy một câu khác để phân tích
text = "Apple is looking at buying U.K. startup for $1 billion"
doc = nlp(text)
```

```
# In ra thông tin của từng token
print(f"{TEXT}<12} | {DEP}<10} | {HEAD TEXT}<12} | {HEAD POS}<8} | {CHILDREN}")
print("-" * 70)

for token in doc:
    # Trích xuất các thuộc tính
    children = [child.text for child in token.children]
    print(f"{token.text}<12} | {token.dep_}<10} | {token.head.text}<12} | {token.head.pos_}<8} | {children}")
```

TEXT	DEP	HEAD TEXT	HEAD POS	CHILDREN
Apple	nsubj	looking	VERB	[]
is	aux	looking	VERB	[]
looking	ROOT	looking	VERB	['Apple', 'is', 'at']
at	prep	looking	VERB	['buying']
buying	pcomp	at	ADP	['startup']
U.K.	compound	startup	NOUN	[]
startup	dobj	buying	VERB	['for']

for	prep	startup	NOUN	['billion']
\$	quantmod	billion	NUM	[]
1	compound	billion	NUM	[]
billion	pobj	for	ADP	['\$', '1']

✓ Phần 4: Duyệt cây phụ thuộc để trích xuất thông tin

4.1. Bài toán: Tìm chủ ngữ và tân ngữ của một động từ

```
text = "The cat chased the mouse and the dog watched them."
doc = nlp(text)
```

```
for token in doc:
    # Chỉ tìm các động từ
    if token.pos_ == "VERB":
        verb = token.text
        subject = ""
        obj = ""

    # Tìm chủ ngữ (nsubj) và tân ngữ (dobj) trong các con của động từ
    for child in token.children:
        if child.dep_ == "nsubj":
            subject = child.text
        if child.dep_ == "dobj":
            obj = child.text
    if subject and obj:
        print(f"Found Triplet: ({subject}, {verb}, {obj})")
```

```
Found Triplet: (cat, chased, mouse)
Found Triplet: (dog, watched, them)
```

4.2. Bài toán: Tìm các tính từ bổ nghĩa cho một danh từ

```
text = "The big, fluffy white cat is sleeping on the warm mat."
doc = nlp(text)
```

```
for token in doc:
    # Chỉ tìm các danh từ
    if token.pos_ == "NOUN":
        adjectives = []
        # Tìm các tính từ bổ nghĩa (amod) trong các con của danh từ
        for child in token.children:
            if child.dep_ == "amod":
                adjectives.append(child.text)

        if adjectives:
            print(f"Danh từ '{token.text}' được bổ nghĩa bởi các tính từ: {adjectives}")
```

```
Danh từ 'cat' được bổ nghĩa bởi các tính từ: ['big', 'fluffy', 'white']
Danh từ 'mat' được bổ nghĩa bởi các tính từ: ['warm']
```

✓ Phần 5: Bài tập tự luyện

Bài 1: Tìm động từ chính của câu

```
def find_main_verb(doc):
    for token in doc:
        if token.dep_ == "ROOT" and token.pos_ in ("VERB", "AUX"):
            return token

    for token in doc:
        if token.dep_ == "ROOT":
            return token
    return None
```

```
text = "The quick brown fox jumps over the lazy dog."
doc = nlp(text)
main_verb = find_main_verb(doc)
print("Câu:", text)
print("Động từ chính:", main_verb, "- POS:", main_verb.pos_, "- DEP:", main_verb.dep_)
```

Câu: The quick brown fox jumps over the lazy dog.
Động từ chính: jumps - POS: VERB - DEP: ROOT

✓ Bài 2: Trích xuất các cụm danh từ (Noun Chunks)

```
MODIFIER_DEPS = {"det", "amod", "compound", "poss", "nummod"}

def build_noun_chunk(token):
    left_index = token.i
    right_index = token.i

    for left in token.lefts:
        if left.dep_ in MODIFIER_DEPS:
            left_index = min(left_index, left.i)

    for right in token.rights:
        if right.dep_ in MODIFIER_DEPS:
            right_index = max(right_index, right.i)

    return token.doc[left_index : right_index + 1]

def custom_noun_chunks(doc):
    chunks = []
    for token in doc:
        if token.pos_ in ("NOUN", "PROPN", "PRON") and token.dep_ not in ("compound", "amod"):
            span = build_noun_chunk(token)
            chunks.append(span)
    return chunks
```

```
text = "The big, fluffy white cat is sleeping on the warm mat in the living room."
doc = nlp(text)
print("Câu:", text)
print("Các cụm danh từ (tự trích xuất):")
for span in custom_noun_chunks(doc):
    print("-", span.text)

print("Các cụm danh từ (spaCy noun_chunks):")
for chunk in doc.noun_chunks:
    print("-", chunk.text)
```

Câu: The big, fluffy white cat is sleeping on the warm mat in the living room.
Các cụm danh từ (tự trích xuất):
- The big, fluffy white cat
- the warm mat
- the living room
Các cụm danh từ (spaCy noun_chunks):
- The big, fluffy white cat
- the warm mat
- the living room

✓ Bài 3: Tìm đường đi ngắn nhất trong cây

```
def get_path_to_root(token):
    path = []
    current = token
    while True:
        path.append(current)
        if current.head == current:
            break
        current = current.head
    return path
```

```
text = "The quick brown fox jumps over the lazy dog."
doc = nlp(text)

for token in doc:
    if token.text == "lazy":
        target = token
        break

print("Câu:", text)
print("Token bắt đầu:", target.text)
print("Đường đi đến ROOT:")
path = get_path_to_root(target)
print(" -> ".join([t.text for t in path]))
```

Câu: The quick brown fox jumps over the lazy dog.
Token bắt đầu: lazy
Đường đi đến ROOT:
lazy -> dog -> over -> jumps