Lab 01: Python Dictionary

Storing Disease Information for RAG Systems

CSI403 - Full Stack Program Development

Today's Agenda

Learning:

- What is a Dictionary?
- Why use Dictionary in RAG?
- Creating Dictionaries
- Accessing & Modifying Data
- List of Dictionaries

Time:

- Lecture: 30 min
- Tutorial: 60 min
- Break: 15 min
- Exercise: 45 min
- Submit: 15 min
- Q&A: 15 min

What is a Dictionary?

A Dictionary stores data as KEY-VALUE pairs

Real Dictionary:

- Word → Definition
- "Apple" → "A fruit"
- "Python" → "A programming language"

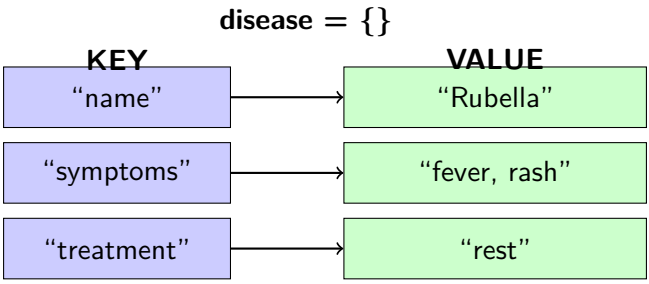
Python Dictionary:

- Key → Value
- "name" → "Rubella"
- "symptoms" → "fever, rash"

Key Point

Use **keys** to quickly find **values** - like looking up a word!

Dictionary Visualization



Why Use Dictionary in RAG Systems?

RAG = Retrieval-Augmented Generation

In RAG systems, we need to store **documents** with **metadata**:

```
document = {
    "title": "Rubella",
    "content": "Rubella is...",
    "source": "medthai.com",
    "date": "2024-01-15"
}
```

Benefits:

- Organized structure
- Fast key lookup
- Easy to extend
- JSON compatible

Important

Almost all RAG systems use dictionaries to store documents!

Creating a Dictionary

Method 1: Empty dictionary, then add

```
# Create empty dictionary
disease = {}

# Add key-value pairs
disease["name"] = "Rubella"
disease["symptoms"] = "fever, rash"
```

Method 2: Create with data

```
# Create with initial data
disease = {
    "name": "Rubella",
    "symptoms": "fever, rash",
    "treatment": "rest"
}
```

Accessing Dictionary Values

Using square brackets [ ]

```
disease = {
    "name": "Rubella",
    "symptoms": "fever, rash"
}

# Access by key
print(disease["name"])        # Output: Rubella
print(disease["symptoms"])    # Output: fever, rash
```

Using .get() method (safer)

```
# Returns None if key doesn't exist
print(disease.get("treatment"))         # Output: None
print(disease.get("treatment", "N/A")) # Output: N/A
```

Adding & Modifying Values

```
disease = {"name": "Rubella", "symptoms": "fever"}

# ADD a new key
disease["treatment"] = "rest"
disease["prevention"] = "vaccination"

# MODIFY existing value
disease["symptoms"] = "fever, rash, fatigue"

# DELETE a key
del disease["prevention"]

print(disease)
# {'name': 'Rubella', 'symptoms': 'fever, rash,
#  fatigue', 'treatment': 'rest'}
```

List of Dictionaries

Store multiple items using a list:

```
diseases = [
    {"name": "Rubella", "symptoms": "fever, rash"},
    {"name": "Cholera", "symptoms": "diarrhea"},
    {"name": "GERD", "symptoms": "heartburn"}
]

# Access first disease
print(diseases[0]["name"])  # Output: Rubella

# Access second disease symptoms
print(diseases[1]["symptoms"])  # Output: diarrhea
```

This is how RAG systems store documents!

Each document = 1 dictionary in a list

Looping Through Data

Loop through list of dictionaries:

```
diseases = [
    {"name": "Rubella", "symptoms": "fever, rash"},
    {"name": "Cholera", "symptoms": "diarrhea"},
    {"name": "GERD", "symptoms": "heartburn"}
]

# Display all disease names
for disease in diseases:
    print(disease["name"])

# Output:
# Rubella
# Cholera
# GERD
```

Useful Dictionary Methods

```
disease = {"name": "Rubella", "symptoms": "fever"}

# Get all keys
print(disease.keys())   # dict_keys(['name', 'symptoms'])

# Get all values
print(disease.values()) # dict_values(['Rubella', 'fever'])

# Get all key-value pairs
print(disease.items())  # dict_items([...])

# Check if key exists
print("name" in disease)       # True
print("treatment" in disease)  # False
```

Summary

What we learned:

- Dictionary = key-value pairs
- Create: `{}` or `dict()`
- Access: `d[``key'']` or `d.get(``key'')`
- Add/Modify: `d[``key''] = value`
- Delete: `del d[``key'']`
- List of dicts for multiple items

Next steps:

1. Complete Tutorial Notebook
2. Do 5 Exercises
3. Push to GitHub
4. Check auto-grading score

Remember

Dictionaries are the foundation of storing documents in RAG systems!

Exercise Preview (5 exercises, 100 points)

1. **Create a dictionary** (20 pts)
   Create a dictionary for "Dengue Fever" disease
2. **Add a new key** (20 pts)
   Add "prevention" key to your dictionary
3. **Access values** (20 pts)
   Print the symptoms of the disease
4. **Create list of dictionaries** (20 pts)
   Create a list containing 3 diseases
5. **Loop through data** (20 pts)
   Loop and print all disease names

Time: 45 minutes

Work on `exercise/Lab01_Exercise.ipynb`

Questions?

Let's start the Tutorial!

Open: `tutorial/Lab01_Tutorial.ipynb`