JSP3: ChainNet Analysis

Integrated Extraction of Metaphor and Metonymy

Vojtěch Mahdal & Lea Karolína Komárková
vmahdal05@gmail.com, leakom11@gmail.com

February 2026

Abstract

Abstract: This report details the extraction and integration of figurative language data from ChainNet. Specifically, we merge and analyze two distinct datasets: *ChainNet-Metaphor* and *ChainNet-Metonymy*. The objective is to unify these resources into a single computational framework, allowing for a direct comparison of lexical shifts across both figurative types. We provide a Python-based methodology for extracting, tagging, and combining these entries into a unified dataset.

---

1 Introduction

The ChainNet project provides granular data on how word meanings shift via cognitive mechanisms. These shifts are primarily categorized into two types:

- **Metaphor:** A mapping between two different conceptual domains (e.g., *Time is Money*).
- **Metonymy:** A mapping within a single conceptual domain (e.g., *The Crown* standing for the Monarch).

Our task was to process the raw JSON data for both mechanisms and combine them into a unified dataset for analysis.

2 Data Structure

The source data consists of two JSON files:

1. `chainnet_metaphor.json`: Contains mappings where the sense shift crosses domain boundaries.
2. `chainnet_metonymy.json`: Contains mappings where the sense shift is contiguous/associative.

Both files follow a specific schema using `from_sense` (source) and `to_sense` (target) identifiers linked to WordNet.

3 Methodology Code

We developed a Python script to ingest both files simultaneously. The script performs the following operations:

1. **Path Auto-Correction:** Detects files even if extensions are hidden or doubled.
2. **Data Normalization:** Extracts the relevant fields (`wordform`, `from_sense`, `to_sense`) from the JSON structure.
3. **Tagging:** Adds a `Type` column to distinguish between Metaphor and Metonymy in the final output.
4. **Aggregation:** Merges both streams into a single CSV dataset.

Listing 1: Python Script for Integrating ChainNet Data

```python
import pandas as pd
import json
import os

# --- 1. CONFIGURATION ---
# Define the input files and their corresponding labels
FILES_TO_LOAD = [
    {"path": r"C:\Users\Mahdal\Desktop\Bond\chainnet_metaphor.json", "type": "Metaphor"},
    {"path": r"C:\Users\Mahdal\Desktop\Bond\chainnet_metonymy.json", "type": "Metonymy"}
]

OUTPUT_FILE = "chainnet_combined_results.csv"

# --- 2. DATA LOADING & PROCESSING ---

def load_and_combine_chainnet(files_config):
    """
    Reads multiple ChainNet JSON files and combines them into one dataset.
    """
    combined_data = []

    for config in files_config:
        filepath = config["path"]
        data_type = config["type"]

        # Robust path checking (handles .json.json errors)
        if not os.path.exists(filepath):
            if os.path.exists(filepath + ".json"):
                filepath += ".json"
            else:
                print(f"Warning: File not found: {filepath}")
                continue

        print(f"Processing {data_type} from {os.path.basename(filepath)}...")

        try:
            with open(filepath, 'r', encoding='utf-8') as f:
                content = json.load(f)

            # Extract the list of entries from the "content" key
            items = content.get('content', [])

            for item in items:
                # Create a clean entry row
```

```
                entry = {
                    'Word': item.get('wordform', 'N/A'),
                    'Type': data_type,
                    'Source_Sense': item.get('from_sense', 'N/A'),
                    'Target_Sense': item.get('to_sense', 'N/A')
                }
                combined_data.append(entry)

            print(f"   -> Loaded {len(items)} entries.")

        except Exception as e:
            print(f"   -> Error reading file: {e}")

    return combined_data

# --- 3. EXECUTION ---

if __name__ == "__main__":
    print("--- STARTING CHAINNET INTEGRATION ---")

    # 1. Load and Combine
    all_data = load_and_combine_chainnet(FILES_TO_LOAD)

    # 2. Output Results
    if all_data:
        df = pd.DataFrame(all_data)

        # Display large sample to console (50 items)
        print("\n--- COMBINED DATASET SAMPLE ---")
        print(df.head(50).to_string(index=False))

        # Save to CSV
        output_dir = os.path.dirname(FILES_TO_LOAD[0]["path"])
        output_path = os.path.join(output_dir, OUTPUT_FILE)

        df.to_csv(output_path, index=False)
        print(f"\nSUCCESS: Saved {len(df)} rows to '{output_path}'")
    else:
        print("\nNo data found. Please check your JSON file locations.")
```

4 Results: Extended Dataset

The table below presents a selection of 50 entries extracted from the combined dataset, illustrating the variety of metaphorical and metonymic shifts preserved in ChainNet.

Table 1: Integrated ChainNet Data (50 Examples)

Word	Type	Source Sense	Target Sense
anchor	Metaphor	anchor%1:06:00::	anchor%1:18:00::
apron	Metaphor	apron%1:06:00::	apron%1:15:00::
bangle	Metaphor	bangle%1:06:01::	bangle%1:06:00::
brake	Metonymy	brake%1:20:02::	brake%1:14:00::
buckeye	Metonymy	buckeye%1:20:00::	buckeye%1:18:00::
cloud	Metaphor	cloud%1:17:00::	cloud%1:26:00::
cockpit	Metaphor	cockpit%1:06:00::	cockpit%1:06:01::

Word	Type	Source Sense	Target Sense
delivery	Metaphor	delivery%1:04:04::	delivery%1:04:03::
difficulty	Metonymy	difficulty%1:09:02::	difficulty%1:07:00::
disappointment	Metonymy	disappointment%1:12:00::	disappointment%1:04:00::
draftsman	Metaphor	draftsman%1:18:00::	draftsman%1:18:01::
edition	Metonymy	edition%1:10:02::	edition%1:14:00::
eighties	Metaphor	eighties%1:28:00::	eighties%1:28:02::
elevation	Metaphor	elevation%1:24:00::	elevation%1:11:00::
escutcheon	Metaphor	escutcheon%1:06:00::	escutcheon%1:06:02::
hatchet	Metaphor	hatchet%1:06:00::	hatchet%1:06:01::
hold	Metaphor	hold%1:04:00::	hold%1:06:02::
identity	Metonymy	identity%1:07:02::	identity%1:24:01::
javelin	Metonymy	javelin%1:06:00::	javelin%1:11:00::
jihad	Metaphor	jihad%1:04:01::	jihad%1:04:00::
kink	Metaphor	kink%1:25:00::	kink%1:09:01::
mil	Metaphor	mil%1:23:02::	mil%1:23:05::
money	Metonymy	money%1:21:01::	money%1:21:00::
monitor	Metaphor	monitor%1:18:01::	monitor%1:06:00::
mystery	Metonymy	mystery%1:09:00::	mystery%1:10:00::
note	Metaphor	note%1:10:04::	note%1:10:05::
octave	Metaphor	octave%1:10:00::	octave%1:28:00::
ossification	Metaphor	ossification%1:22:00::	ossification%1:22:02::
potato	Metonymy	potato%1:13:00::	potato%1:20:00::
profundity	Metonymy	profundity%1:07:01::	profundity%1:09:01::
queen	Metaphor	queen%1:18:03::	queen%1:18:04::
receiver	Metonymy	receiver%1:18:00::	receiver%1:18:02::
recital	Metonymy	recital%1:10:04::	recital%1:10:01::
reserve	Metonymy	reserve%1:21:00::	reserve%1:14:00::
rhetoric	Metaphor	rhetoric%1:10:00::	rhetoric%1:10:02::
separation	Metonymy	separation%1:11:00::	separation%1:15:00::
sequel	Metaphor	sequel%1:10:00::	sequel%1:11:00::
shovel	Metonymy	shovel%1:06:00::	shovel%1:23:00::
silver	Metonymy	silver%1:27:00::	silver%1:06:01::
sitting	Metonymy	sitting%1:04:00::	sitting%1:14:00::
smell	Metonymy	smell%1:07:00::	smell%1:09:02::
stretch	Metonymy	stretch%1:04:01::	stretch%1:04:00::
string	Metonymy	string%1:06:00::	string%1:06:03::
sucker	Metonymy	sucker%1:18:00::	sucker%1:05:02::
suggestion	Metaphor	suggestion%1:10:00::	suggestion%1:10:01::
tea	Metonymy	tea%1:13:00::	tea%1:20:00::
throttle	Metonymy	throttle%1:06:01::	throttle%1:06:00::
title	Metaphor	title%1:10:00::	title%1:10:02::
traffic	Metaphor	traffic%1:10:00::	traffic%1:04:02::

Word	Type	Source Sense	Target Sense
zoom	Metonymy	zoom%1:04:00::	zoom%1:11:00::

5 Discussion

By excluding external conceptual categories (MML) and focusing purely on the internal structure of ChainNet, we reveal the specific "sense-to-sense" architecture of the database.

Notice how words like **"money"** and **"hold"** appear in the dataset with distinct cognitive shifts:

- **Metonymy (Money):** Shifts from the abstract currency (money%1:21:01) to the specific medium of exchange (money%1:21:00).
- **Metaphor (Hold):** Shifts from the physical act (hold%1:04:00) to a container/storage sense (hold%1:06:02).

This integrated view confirms that polysemy in ChainNet is modeled not just as a random collection of meanings, but as a structured network of cognitive shifts.

6 Conclusions

We have successfully extracted and merged the metaphor and metonymy datasets. The resulting CSV file serves as a foundation for training classifiers that can distinguish between these two fundamental types of figurative language based solely on WordNet sense keys.