CS – 6320 : Natural Language Processing

Project : Information extraction Application using NLP features and Techniques

Team Name : LoneWolf

A. Problem description:

Information Extraction is one of the important tasks in the field of Natural Language Processing (NLP). It has a wide range of applications such as Question Answering systems, Text Summarization, Event Extraction, relation extraction, etc. In this project, Information Extraction is done using Template filling on unstructured text data from Wikipedia. This project aims to use NLP features and stack NLP techniques in a pipeline to extract the following template from the given sentence.

Input:

- 10 articles related to Organizations
- 10 articles related to Persons
- 10 articles related to Locations

Templates:

- Template #1:

  BORN(Person/Organization, Date, Location)

- Template #2:

  ACQUIRE(Organization, Organization, Date)

  - Organization in argument1 acquired the Organization motioned in argument2

- Template #3:
  PART_OF(Organization, Organization)
  PART_OF(Location, Location)

B. Proposed solution:

Information extraction task is generally solved using multiple approaches. The most common ones are listed below:

- Supervised Approach
- Semi-supervised Approach
- Rule-based Approach

In this project, we will be using Rule-based Approach as we don't have any labeled data. This approach uses the syntactic and grammatical rules followed by the language of the input text to identify the parameters of the template. NLP features commonly used in rule-based approach are listed below:

- Lemmas
- POS tags
- Dependency Parsing
- Named-Entity Recognition
- Pattern Matcher
- Wordnet features – Hypernyms, Hyponyms, Holonyms, Meronyms

<u>GENERAL APPROACH TO THE SOLUTION</u>

- BORN Template
  - Perform named entity recognition to check if the sentence has a 'PERSON' or 'ORG'. Need not look for 'Date' and 'Location' as they are not mandatory for the template
  - Look for words like born, found, started, created in the verbs in the sentence

- Use dependency parse to make sure the right ‘PERSON’ or ‘ORG’ linked to the trigger word is filled in the template

- ACQUIRE Template
  - Perform named entity recognition to check if the sentence has two ‘ORG’ in it, as we are looking for acquisition act between two companies. Need not look for ‘Date’, it is optional
  - Look for words like acquire, buy, sell in the verbs in the sentence
  - Check if it is a buying act or selling act and make sure the organizations are filled in the correct order in the template. Dependency parse constraint for both will be different
  - Use dependency parse to make sure the right ‘ORG’ linked to the trigger word is filled in the template

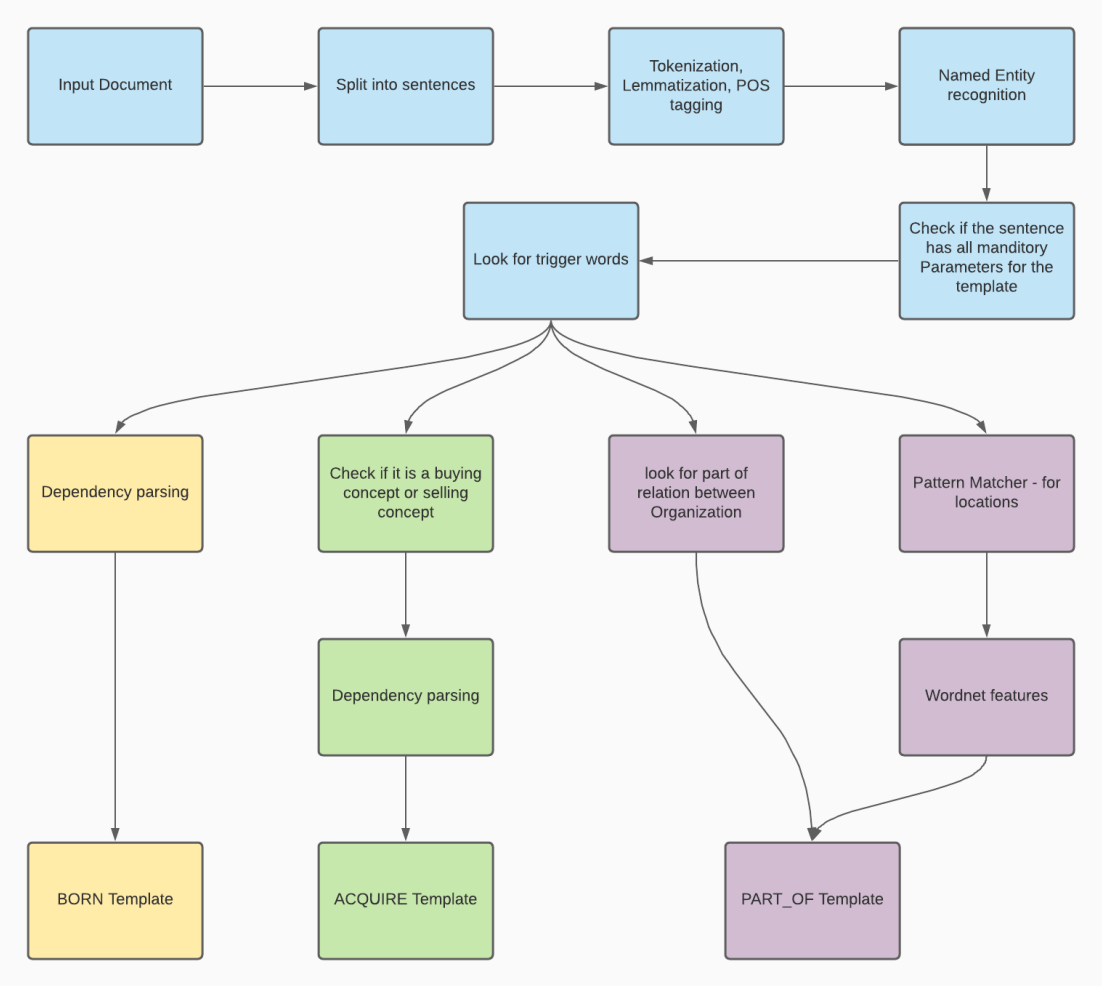
- PART_OF Template
  - Perform named entity recognition to check if the sentence has two ‘ORG’ or two ‘GPE’ in it
  - Look trigger works like ‘part of’ for Organizations.
  - Look for patterns like GPE, GPE for Location
  - Use holonyms or meronyms to make sure there is a hierarchy between the locations.

C. Full implementation details

a. Programming tools

The system is implemented using Python as programming language. NLP libraries used are NLTK and spacy

b. Architectural diagram



The above diagram shows the architecture of the NLP pipeline used for Template extraction

c. Results and error analysis

- Spacy NER was not able to recognize a few words:
  - In May 2002, Musk founded SpaceX, an aerospace manufacturer and space transport services company, of which he is CEO and lead designer.
    The model was not able to recognize SpaceX as 'ORG'

- Spacy NER incorrectly recognize a few words:
  - Lincoln grew up on the frontier in a poor family.
    The model incorrectly recognized Lincoln as 'ORG'

- Wordnet features like holonyms and meronyms does not have fine information of location like cities and towns:
    - Can not use it to check for hierarchy for those small locations

d. A summary of the problems encountered during the project and how these issues were resolved

- Order of organizations is important in ACQUIRE Template.
  Solution: split the task into buying concept and selling concept
- All parameters of the template are Named Entities, So most of the processing is done with Named Entities. It was difficult to verify it with Dependency parse as it breaks the Named entity into subject and compound or subject and modifier (for DATES)
  Eg: Whole Foods Market is broken down into 3 words
  Solution: Instead of just iterating through Named entity, have another loop to iterate through words in each and check for constraints on the head of the Named entity, if satisfied, include the Named entity to the template
- Neuralcoref package in Spacy is not compatible with python 3.8

e. Pending issues

- Not able to recognize if a sentence has the same template twice
    - "I live in Richardson, Texas, US."
    - "Nepal is located in Asia, and it was part of India before the 1980s."
    - "Amazon's acquisitions include eBay, Facebook, and Whole Foods Market."

f. Potential improvements

- Include Coreference resolution to learn relation involving pronoun
- Instead of using default model for NER, we can train a model to improve its performance or can add rules to the existing model.
- Look for more false Positives and false Negatives of our model and look for ways to handle it