Project Description

Implementing an Information Extraction application using NLP features and techniques:

Stage 1: Input Data reading:

30 text articles:

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

Stage 2: Information extraction templates

- Template #1: *BUY (Buyer, Item, Price, Quantity, Source)*
- Template #2: *WORK (Person, Organization, Position, Location)*
- Template #3: *PART (Location, Location)*

Stage 3: Implementation of NLP techniques to extract NLP features

- Tokenization
- Lemmatization
- Part-Of-Speech Tagging
- Dependency Parsing
- Word Relations: Hypernyms, Hyponyms, Holonyms, Meronyms

Stage 4: Implementation of a heuristic-based approach to extract filled information templates from the corpus

Proposed Solution

Programming Language: Python 3.7

Open Source Libraries: NLTK, Spacy to extract NLP features

Generate Heuristics for each template by extracting NLP features and Named Entity Recognition is performed for Template Matching and Template Filling

Architecture

1. Co-reference resolution for the given file
2. Heuristic Generation for Each Template
3. Sentence reading for Each Template
4. Tokenization
5. Word Lemmatization
6. Part of Speech Tagging
7. Dependency Parsing
8. Hypernyms, Holonyms, Hyponyms, and Meronyms
9. Sentence Template Matching
10. Identifying Template Attributes using Heuristics
11. Template filling
12. Generation of JSON output file.

Assumptions

1. A sentence can fill multiple templates
2. Multiple sentences can fill the same template, but they must be contiguous

Summary of Problems encountered

1. There were some inaccurate Named Entity Recognition by Spacy. E.g. Richardson was identified as a PROPN rather than GPE,

Solution: Loaded a dataset for most popular cities, states, and countries. This helps to classify Richardson as a city.

2. Structural ambiguity in the Parse tree lead to ambiguity in identifying some components of the sentence

3. When the structure of the sentence differed as Passive or Active, some Verbs were identified as Nouns that lead to the problem in identifying objects. E.g. Location

Solution: Used words like “by” to differentiate between Active and Passive voice. And then appropriately trigger those rules for Active voice or Passive voice

4. For the PART (Location, Location) template, we faced a few challenges identifying and properly extracting the Locations from the sentence.

Solution: The issue was resolved by generating a different set of heuristics for different occurrence case of locations in a sentence

Future Scope

1. Accuracy was calculated by separately labeling (classifying) each sentence of the Wikipedia articles as BUY, WORK, PART, None. A separate program was written for this which can be improved and made more user-friendly.
2. The structural ambiguities in a sentence can be more efficiently handled by generating more sophisticated heuristics.
3. Deep learning can be used to handle multiple templates.
4. Current co-referencing didn’t improve the performance significantly, Hence, to develop a better and more robust co-referencing, which would improve the performance.