Project Report

Natural Language Processing (CS6320) - Fall 2018

Harsh Bhalani (hpb170030) & Harsh Bhakt (hbb170030)

Definition:

Project is to build a template based Information Extraction application using NLP features and techniques. Project required following steps.

1. Decide Domain: Movie Information
2. Corpus Collection: Corpus was collected from authentic movie sources. i.e. Wikipedia.
3. Template Extraction: 11 templates were created with 40+ properties.
4. Feature extraction pipelines: Implemented pipelines to extract following features-

    Sentence Tokenization
    Word Tokenization
    POS Tagging
    lemma
    Dependency
    Hypernym
    Holonym
    Meronym
    Hyponym
    Synonym

5. Template filling: Discussed in reporting.

Templates:

1. **Write (product, writer, year)**
    - **Example:** The Searching is directed by Aneesh in his feature debut and written by Chaganty, Ohanian and John in 2016.
    - **Output:**
        - Product: The Searching
        - Writer: “Chaganty”, “Ohanian”, “John”
        - Year: 2016
2. **Buy (Buyer, When, Seller, What, Amount, where)**
    - **Example:** The rights of Iron Man were sold to Universal Studios for by Fox Media for $3,000,000 in May, 2015 at Preston Theater.
    - **Output:**
        - Buyer: Universal Studiod
        - When: May, 2015
        - Seller: Fox Media
        - What: The rights
        - Amount: $3,000,000
        - Where: Preston Theater
3. **MovieEarning (Movie, Amount, Location)**

- **Example:** As of December 2, 2018, First Man has grossed $44.7 million in the United States, and $55.2 million in other territories, for a total worldwide gross of $99.9 million, against a production budget of $59 million.
- **Output:**
  - Movie: First Man
  - Amount: $44.7 million
  - Location: United States

4. **PlayedRole(person, role, time)**
   - **Example:** The role of Gillian B. Loeb was played by Colin McFarlane.
   - **Output:**
     - Person: Colin McFarlane
     - Role: GIllian B. Loeb
     - Time: <no-match>
5. **PromotionalEvent (Movie, Event, Location, Date)**
   - **Example:** The premiere of IronMan was held in the Greater Union Theater at George Street, Sydney, on April 14, 2008.
   - **Output:**
     - Movie: IronMan
     - Event: premiere
     - Location: Greater Union Theater
     - Date: April 14, 2008
6. **Release (product, location, date, owner, format)**
   - **Example:** Black Panther premiered in Los Angeles on January 29, 2018, and was released theatrically in the United States on February 16, in 2D, 3D, IMAX and other premium large formats.
   - **Output:**
     - Product: Black Panther
     - Location: Los Angeles
     - Date: January 29, 2018
     - Owner: <no-match>
     - Format: 2D, 3D, IMAX
7. **MovieRating (movie, rating, rateBy, time, totalReviews)**
   - **Example:** Rotten Tomatoes gave 7/10 ratings to the Mission Impossible in 2015 based on 273 reviews .
   - **Output:**
     - Movie:The Mission Impossible
     - Rating: 7/10
     - RateBy: Rotten Tomatoes
     - Time: 2015
     - TotalReviews: 273

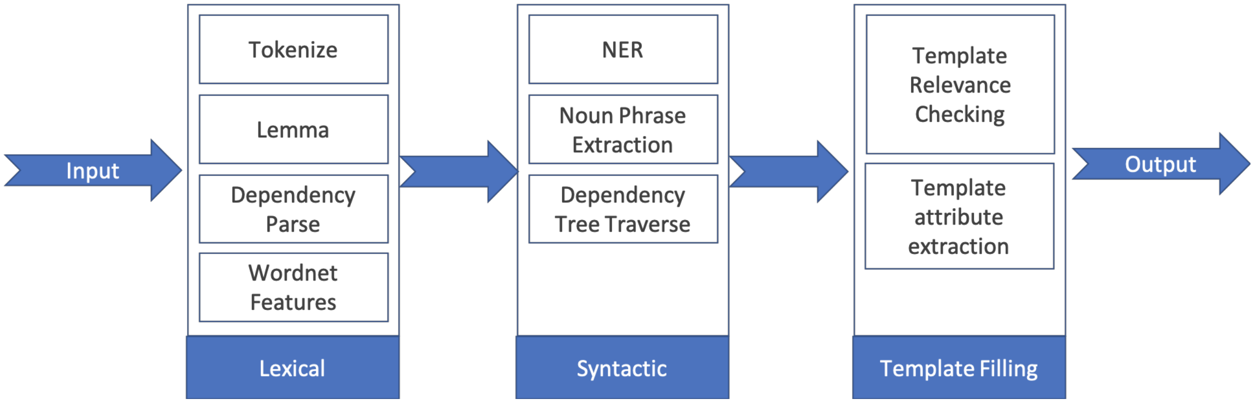
8. **RecognizedAs (movie, recognizer, recognizedAs, typeOfRecognition, time)**

    - **Example:** The Angry Man was awarded the second-best courtroom drama ever by the American Film Institute in 2010.
    - **Output:**
        - Movie: The Angry Man
        - Recognizer: The American Film Institute
        - RecognizedAs: the second-best courtroom drama
        - TypeOfRecognition: Win
        - Time: 2010

9. **Filming(Movie/scene, Location, time)**
    - **Example:** On April 10, 2017, filming of Fallout was slated to start in Paris.
    - **Output:**
        - Movie/Scene: FallOut
        - Location: Paris
        - Time: April 10, 2017
10. **Viewed(movieName, numberOfTimes, duration)**
    - **Example:** The trailer of The Lion King was viewed 224.6 million times in its first 24 hours.
    - **Output:**
        - MovieName: The Lion King
        - NumberOfTimes: 224.6 million
        - Duration: 24 hours
11. **Remake(remakeMovie, originalMovie, Year)**
    - **Example:** The Magnificent Seven is a remake movie of Seven Samurai created in 2016.
    - **Output:**
        - RemakeMovie: The Magnificent Seven
        - OriginalMovie: Seven Samurai
        - Year: 2016

Implementation Pipeline:

For NLP Feature extraction, NLTK and SpaCy libraries were used.



Rationale:

- Synonyms is a good way to detect matching sentences. But after analyzing corpus we realized only few synonyms are used in the domain. For example, only “Release” and “Launched” are used to indicate movie release. So we discarded all other relevant synonyms of release and using just a subset of synonyms.
- Instead of using common pipeline for all templates, we used individual pipelines for each template. This allowed us to do template specific modification without worrying about maintaining common pipeline. Also, team members could work on different templates in parallel.

Problem solutions:

- **Single token to multi-word entity detection:**
  - One problem we faced was detecting entire entity or noun phrase from single token.
  - We implemented a function which takes sentence and token.
  - Then it extract entity (NER) / noun phrases.
  - Then matches token’s text in the extracted entity / noun phrases.
- **Date extraction from heuristic to syntactic:**
  - We started with naive approach of detecting DATE using NER for MovieRelease Template.
  - We faced problem when sentence had two dates.
  - One solution was to match child key temporal word text and find DATE.
  - But this did not work when both date’s key temporal words had the same text.

    - E.g. Movie premiered on 2 **January**, 1989 and was released on 3 **January**, 1989.
    - To solve this problem, we figured out a way to detect token position so finally we could find correct date for release and premier.

- **Correction in location extraction method:**
    - After analyzing corpus, we started detecting location information as child of ["at", "in"].
    - The problem was NER was not recognising LOCATIONS correctly. Sometimes TIME also can be attached with “at” and thus can not ensure the Location.
    - We added one extra filter for “at” children.
    - Based on token’s position and text we extracted DATE entities. If it returned positive means that child is TIME and not Location.

Pending problems:

- **Imperfect multi-word entity detection:**
    - The function we created to detect token to multi-word entity is not 100% accurate because it relies on NER and Noun Phrase Detector.
- **Movie name extraction issue:**
    - Unable to detect movie names correctly which contains ":" or “-” e.g. "Captain America: The First Avenger" is detected as “The First Avenger”.