Information Extraction

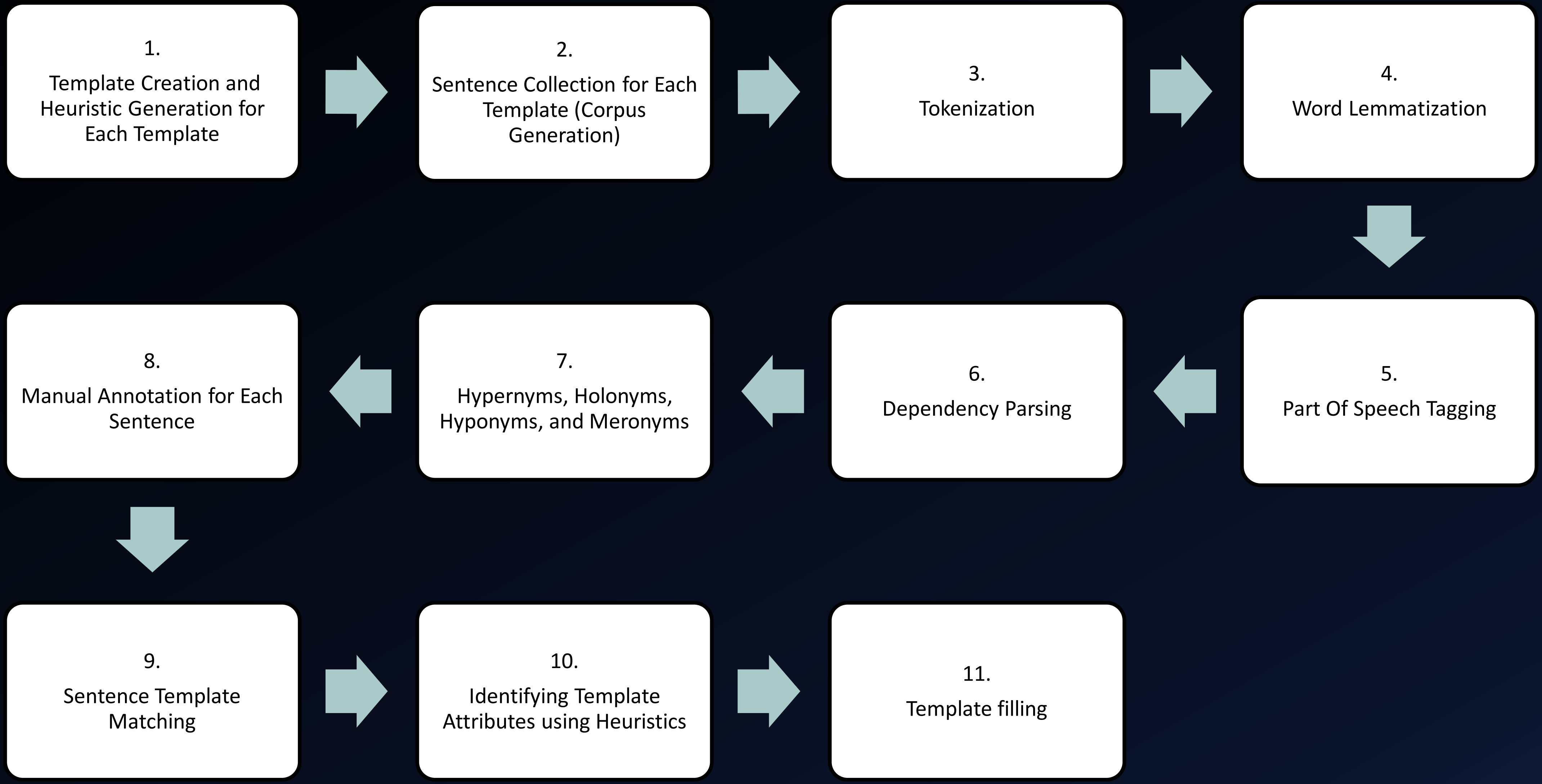
Problem Description

- Implement an Information Extraction application using NLP features
- Project comprised of 4 stages:
  - Stage 1: Creation of at least 10 unique information templates with cumulative 40 attributes
  - Stage 2: Creation of corpus of at least 50,000 words
  - 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 machine-learning, statistical, or heuristic based approach to extract filled information templates from the corpus

PROPOSED SOLUTION

- Selected Domain : Crime
- Programming Language: Python 3.6
- Open Source Libraries: NLTK, Spacy
- Manual Creation of 10 templates with the required properties by exploring various authentic resources for crime reports
- Using text scraping and manual exploration, collect the required corpus
- Using open source libraries such as NLTK and Spacy, extract NLP features
- Generate Heuristics for each template, the extracted NLP features and Named Entity Recognition perform Template Matching and Template Filling

ARCHITECTURAL DIAGRAM



1. Template Creation and Heuristic Generation for Each Template

2. Sentence Collection for Each Template (Corpus Generation)

3. Tokenization

4. Word Lemmatization

5. Part Of Speech Tagging

6. Dependency Parsing

7. Hypernyms, Holonyms, Hyponyms, and Meronyms

8. Manual Annotation for Each Sentence

9. Sentence Template Matching

10. Identifying Template Attributes using Heuristics

11. Template filling

SOME TEMPLATES

1. Murder < Date, Location, Culprit, Victim, Murder Weapon *(Optional)* >
2. Kidnap < Date, Location, Culprit, Victim, Ransom (Optional) >
3. Rob < Date, Location, Culprit, Victim, Stolen Item (Optional) >
4. Attack < Date, Location, Organization, Damage, Attack Weapon (Optional) >

Assumptions:

- In absence of a date and location, there will be default date and location
- A sentence can fill multiple templates
- Multiple sentences can fill a same template, but they must be contiguous

CHALLENGES FACED

- NLTK WordNet Lemmatizer did not lemmatize all the word properly
- Inaccurate Named Entity Recognition by Spacy
- Predicate Ambiguity
- When the structure of the sentence is different:
  - Passive Sentences
  - Verbs occurred as Nouns
  - Object such as Date, Location, Money etc act as a subject

- The issue faced due to change in structure was resolved by generating a different set of heuristics for each case
- Though there were some unresolved, most were solved

FUTURE SCOPE

- Current system produces an accuracy of 72%
- Accuracy calculated using a score metric for the total number of correctly filled templates for given sentences, with the correct attributes

- To challenge the structural changes in a sentence, can generate more sophisticated and complex heuristics
- Use of a deep learning technique to solve this problem
- Can be extended to multiple templates related to Crime