CS 6320 – Natural Language Processing
Fall 2020
Course Project

TEAM NAME: LINGUISTIC MACHINES

TEAM MEMBERS: RAHUL CHOUDHARY PAVALUR BALAKRISHNA (Net ID: RXP190029)

PATTABHI RAMANNA VELLA (Net ID: PXV180022)

PROBLEM DESCRIPTION

- To design and implement models for extracting relations between two named entities in a sentence.
- The model also determines the direction of the relation between the two named entities.
- SEMEVAL dataset is given which is split into two parts – semeval_traint.txt that contains 8000 examples and semeval_test.txt which contains 2717 examples.
- In each example of the dataset, the following annotations are provided:
    - The spans of the two named entities between which the relation holds (which are indicated by delimiters e1 and e2)
    - The relation (and its directionality) that holds between the two entities
- The SEMEVAL dataset examples contain 10 types of relation classes:
    - Cause-Effect
    - Component-Whole
    - Content-Container
    - Entity-Destination
    - Entity-Origin
    - Instrument-Agency
    - Member-Collection
    - Message-Topic
    - Product-Producer
    - Other
- Each of the examples in the dataset have one of these classes assigned along with the direction that the two entities will define that relation.
- The criteria of this project are:
    - To create a corpus reader that is able to read the data files and represent the information in a way such that the model can process it.
    - Implement a deep NLP pipeline to extract NLP based features from the natural language statements.
    - Implement a machine-learning, statistical or heuristic (or combination) based approach to determine the relation and its direction for the test dataset.
    - To evaluate our NLP system on the test dataset.

PROPOSED SOLUTION

Part 1: Performing Task 1 and Task 2:

For part 1 of the project we used the basic NLP features such as:

- Reading the document.
- Splitting the document into sentences using Spacy model.
- Perform pre-processing on the sentences to remove the annotations (delimiters e1 and e2 and the relation of the sentence).
- Extract the two entities, the relation and its direction for each sentence (using regular expressions).
- Tokenization of the sentences using Spacy model.
- Lemmatization using spacy.
- Identifying the part of speech tags using spacy.
- Identifying the dep tags using spacy.
- The lemmas of the tokens using spacy.
- The synsets of the tokens using nltk:
    - hypernymns, hyponyms, meronyms, and holonyms.
- Dependency parsing using Spacy and its visualization.

Part 2: Performing Task 3 and Task 4:

For part 2 of the project:

- We develop a rule-based heuristic model to determine the relations and its direction.
- We initially extract the part of speech (POS) tags of the entities by tokenizing them using Spacy for each sentence and store them.
- We also extract the dep tags of the entities by tokenizing them using Spacy for each sentence and also store them.
- We extracted the relation of each sentence and stored them for each sentence.
- We then defined rules, unique to each class of the relation, depending upon the POS tags of the entities, the lemmas, their root hypernyms, the verbs in between the entities, etc., using Spacy's matcher model
- Based on these defined rules, the sentence will then be classified into a relation class.
- To identify the directionality of the relation, we define additional rules based on the dep tags of the entities and the position of verbs between the entities.
- The directionality is then assigned based on the rules defined.
- The rules for identifying the relation and its directionality were defined by analyzing the semeval_train dataset provided and then identifying all the required NLP features from part 1 of the project.

IMPLEMENTATION DETAILS

Programming Tools:

Spacy:

For:

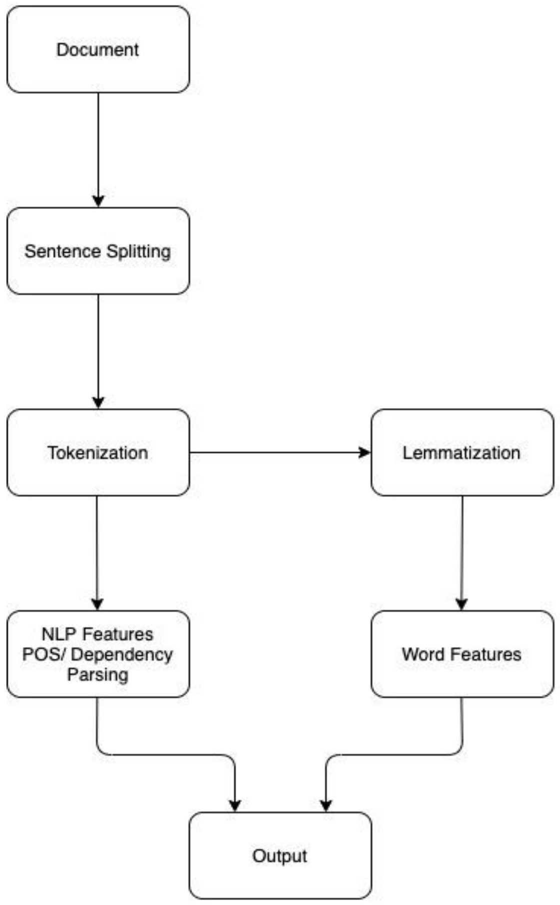
- Sentence segmentation
- Tokenization
- Lemmatization
- pos tagging
- dep tagging
- dependency parsing
- defining the rules
- rule matching

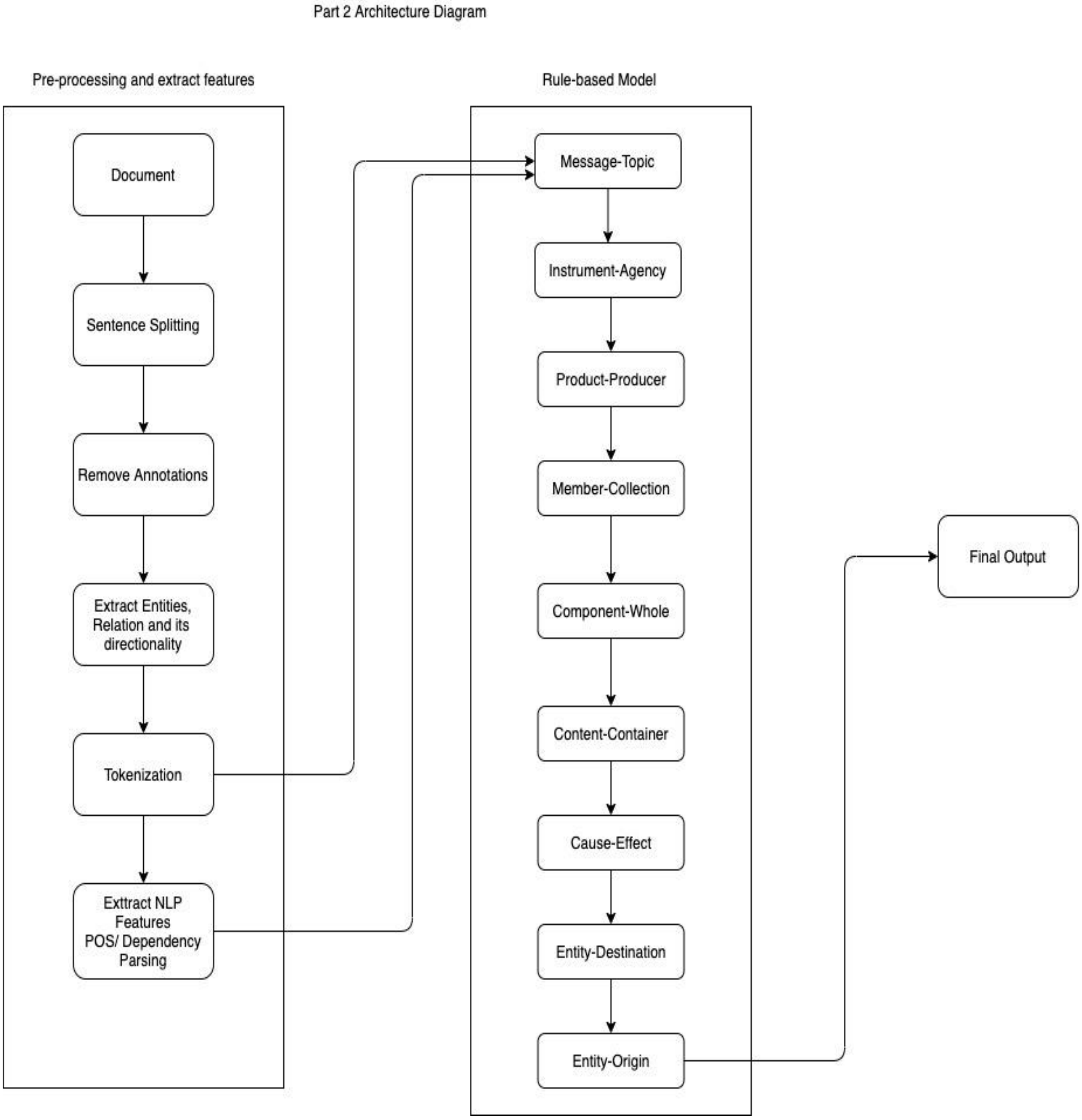
NLTK:

- To find the token/word features (hypernymns, hyponyms, meronyms, and holonyms)

Architecture Diagram:

Part 1 Architecture Diagram





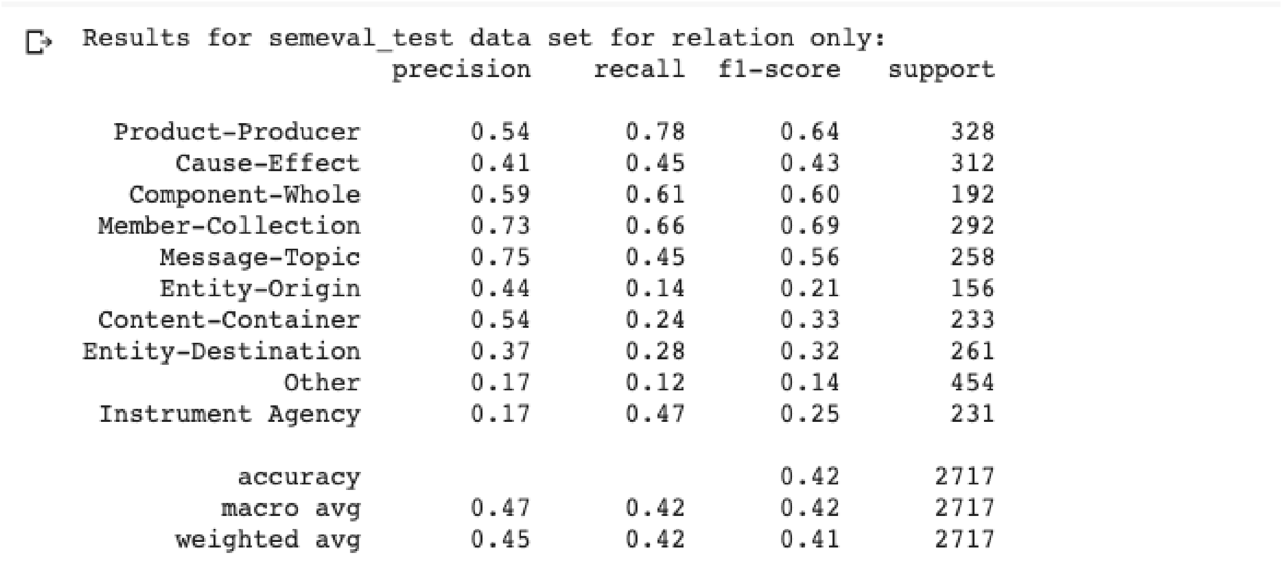
Results and Error Analysis:

Running the heuristic rule-based model on semeval_test dataset, the following results were obtained:

- Setting 1: (Assuming only the relation is classified correctly):

Classification Report:

Accuracy: 42%

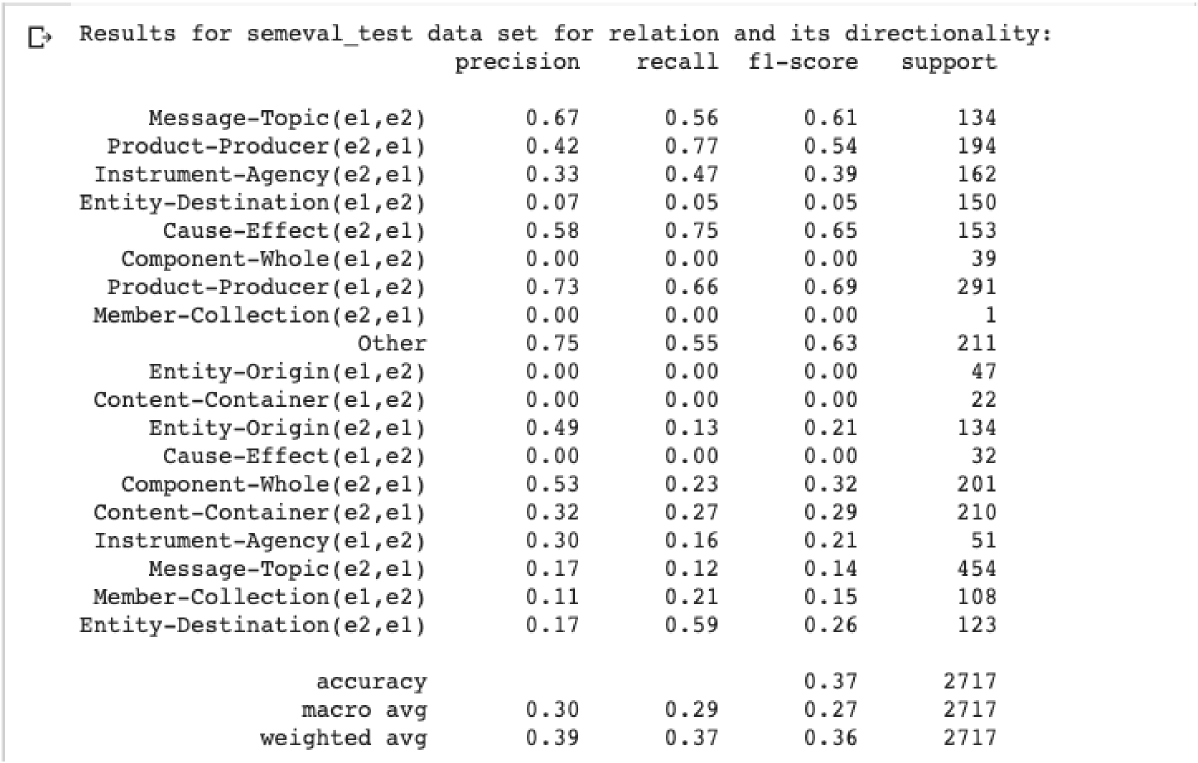


```
Results for semeval_test data set for relation only:
                      precision    recall  f1-score   support

    Product-Producer       0.54      0.78      0.64       328
        Cause-Effect       0.41      0.45      0.43       312
     Component-Whole       0.59      0.61      0.60       192
   Member-Collection       0.73      0.66      0.69       292
       Message-Topic       0.75      0.45      0.56       258
       Entity-Origin       0.44      0.14      0.21       156
   Content-Container       0.54      0.24      0.33       233
  Entity-Destination       0.37      0.28      0.32       261
               Other       0.17      0.12      0.14       454
   Instrument Agency       0.17      0.47      0.25       231

            accuracy                           0.42      2717
           macro avg       0.47      0.42      0.42      2717
        weighted avg       0.45      0.42      0.41      2717
```

- Setting 2: (Assuming both the relation and direction are classified correctly):

Classification Report:

Accuracy: 37%

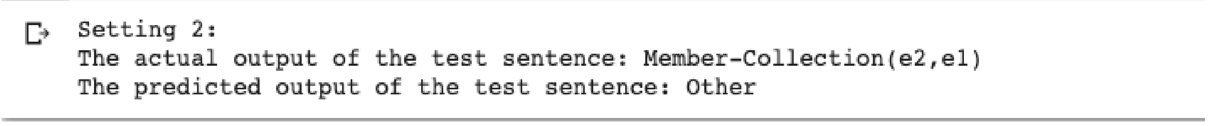


```
Results for semeval_test data set for relation and its directionality:
                          precision    recall  f1-score   support

     Message-Topic(e1,e2)       0.67      0.56      0.61       134
  Product-Producer(e2,e1)       0.42      0.77      0.54       194
  Instrument-Agency(e2,e1)      0.33      0.47      0.39       162
Entity-Destination(e1,e2)       0.07      0.05      0.05       150
      Cause-Effect(e2,e1)       0.58      0.75      0.65       153
   Component-Whole(e1,e2)       0.00      0.00      0.00        39
  Product-Producer(e1,e2)       0.73      0.66      0.69       291
 Member-Collection(e2,e1)       0.00      0.00      0.00         1
                    Other       0.75      0.55      0.63       211
     Entity-Origin(e1,e2)       0.00      0.00      0.00        47
 Content-Container(e1,e2)       0.00      0.00      0.00        22
     Entity-Origin(e2,e1)       0.49      0.13      0.21       134
      Cause-Effect(e1,e2)       0.00      0.00      0.00        32
   Component-Whole(e2,e1)       0.53      0.23      0.32       201
 Content-Container(e2,e1)       0.32      0.27      0.29       210
 Instrument-Agency(e1,e2)       0.30      0.16      0.21        51
     Message-Topic(e2,e1)       0.17      0.12      0.14       454
 Member-Collection(e1,e2)       0.11      0.21      0.15       108
Entity-Destination(e2,e1)       0.17      0.59      0.26       123

                 accuracy                           0.37      2717
                macro avg       0.30      0.29      0.27      2717
             weighted avg       0.39      0.37      0.36      2717
```

- The model took approximately **358 seconds** to run, from reading the semeval_test dataset to predicting the relation and direction for all the sentences.
- Error analysis:

Sentence:

"The Constitutionalist and Republican were arm in arm; and the Quaker and Presbyterian forgot their religious antipathies in this <e1>coalition</e1> of <e2>interests</e2>."
Member-Collection(e2,e1)

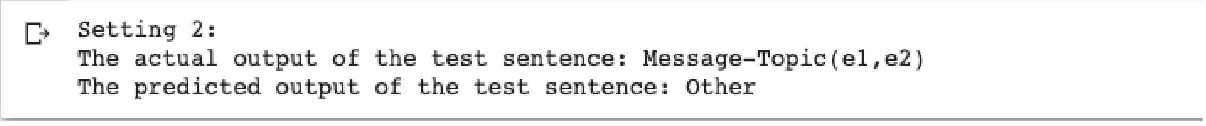


```
Setting 2:
The actual output of the test sentence: Member-Collection(e2,e1)
The predicted output of the test sentence: Other
```

In this example, the rules defined to identify examples of Member-Collection relation class only detect patterns such as collect, of <NOUN>, of <PROPN>, of <ADJ> <NOUN>. But the word between the entities is “of”. A specific rule cannot be made to classify this example, as relations of other classes such as Component-Whole, Content-Container also contain such examples.

Sentence:

"The most common <e1>audits</e1> were about <e2>waste</e2> and recycling."
Message-Topic(e1,e2)



```
Setting 2:
The actual output of the test sentence: Message-Topic(e1,e2)
The predicted output of the test sentence: Other
```

In this example, the rules defined to identify examples of Message-Topic relation class only detect patterns such as <NOUN> .. <VERB> .. <NOUN>. But the word between the entities are <AUX> <ADP>. Hence, this example was not classified into a Message-Topic relation.

Problems Encountered:

- NER tags could not be determined for the entities of the examples in the dataset.
- Specific rules could not be determined for the examples in the dataset for Message-Topic relation class. The examples could not be differentiated from the examples of other classes, even after considering the NLP features such as synset, hypernyms, meronyms, dep tags, pos tags, etc.
- For the examples of Product-Producer relation class, the examples did not follow a specific pattern to differentiate from the rest of the examples. To resolve this problem and identify the examples of this class, we had to identify the verbs in the examples and obtain their hypernyms and root hypernyms. If the root hypernyms

matched any of the words such as make, create, discover, develop, invent, find, manufacture, etc., the example could be classified as Product-Producer.

- The order in which the rules where to be placed to determine the relation class affected the performance of the model. The rules placed toward the end, have better results at classifying the example to their particular relation class. However, some of the examples correctly identified initially, can be affected by rules that are applied later.

Pending Issues:

- NER tags cannot be determined for the entities.
- Specific rules could not be determined for the examples of Message-Topic relation class.
- Rules applied toward the end can wrongly affect correctly classified examples.

Potential Improvements:

- Better derivation of rules by including additional NLP features such as synonyms, hyponyms, etc., to make the rules more sophisticated and customized.
- Improving the model efficiency by providing a way to avoid the correctly classified examples to further be processed by the rules.