Citation Intent Classification

Identifying the Intent of a Citation in scientific papers

Isaac Riley and Pavan Mandava

May 20, 2020

University of Stuttgart
Germany

Task Description

- Identifying intent of a citation in scientific papers
- Three Intent categories/classes from the data set
  1. background (background information)
  2. method (use of methods/tools)
  3. result (comparing results)
- **Classification Task**
  - Assign a discrete class (intent) for each data point

Data set

- Training Data: 8.2K+ data points
  1. background - 4.8K
  2. method - 2.3K
  3. result - 1.1K
- Testing Data: 1.8K data points
  1. background - 1K
  2. method - 0.6K
  3. result - 0.2K

4 lead to a decrease in SC absorption in mice (Deng et al., 2010; Deng et al., 2012).	background
We used an active contour algorithm [10] to segment organs from 340 coronal slices over the two patients.	method
Similar results were found by Sideris et al. (1999) in Greece and Mohebali et al. (2005) in Iran.	result

Table: Sample Dataset

Approach & Architecture

Classifier Implementation

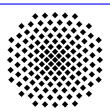
Base Classifier: **Perceptron**

- Linear Classifier
- Binary Classifier

```
class Perceptron:
    def __init__(self, label: str, weights: dict, theta_bias: float)
    def score(self, features: list)
    def update_weights(self, features: list, learning_rate: float, penalize:
    ↪  bool, reward: bool)

class MultiClassPerceptron:
    def __init__(self, epochs: int,learning_rate: float,random_state: int)
    def fit(self, X_train: list, labels: list)
    def predict(self, X_test: list)
```

- **Parameters** and **Hyperparameters**




Approach & Architecture

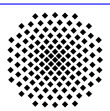
Feature Representation

Lexicons and Regular Expressions ($\approx$ 30 Features)

- LEXICONS

```
ALL_LEXICONS = {
    'INCREASE': ['increase', 'grow', 'intensify', 'build up', 'explode'],
    'USE': ['use', 'using', 'apply', 'applied', 'employ', 'make use'],
    .....
}
```

- REGEX
  - *ACRONYM*
  - *CONTAINS_URL*
  - *ENDS_WITH_ETHYL*



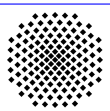

Evaluation of the Classifier
F1 Score

- F1 Score
  - weighted average of Precision and Recall

```
def f1_score(y_true, y_pred, labels, average)
```

- Averaging
  - MACRO
  - MICRO
  - None

- Why MACRO and MICRO ?




Model Performance

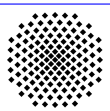
Results

Averaging	Score
MICRO	0.64
MACRO	0.57
background	0.72
method	0.54
result	0.46

Table: F1-Score Results

Next Steps

- Better Feature Representation - Word Embeddings
  - word2vec
  - BERT
  - ELMo
  - ...
- Better Classifier (Non-Linear / Neural Networks)
  - BiRNNs
  - BiLSTMs
  - CNNs
  - ...
- Interaction with other groups




Thanks for listening