# CMPE 493
# Term Project
# Final Presentation

Named Entity Normalization for the Bacteria Biotopes Domain

Şadi Uysal, Enes Özcan, Sercan Ersoy

# Previous Work

- Exact Match
  - Text exactly matches the name of an ontology term
- Jaccard Similarity
  - Jaccard similarity btw every ontology term's name
  - The term with the maximum score is predicted
- Cosine Similarity
  - Cosine similarity btw every ontology term's name
  - The term with the maximum score is predicted

# Previous Results

- Scores for the train data
- Total of 1118 habitats

| | Success | Fail | No Prediction | Accuracy (predicted) | Accuracy (total) |
|---|---|---|---|---|---|
| **Exact Match** | 166 | 43 | 909 | 79,42 | 14,85 |
| **Jaccard Sim** | 323 | 795 | 0 | 28,89 | 28,89 |
| **Cosine Sim** | 341 | 777 | 0 | 30,50 | 30,50 |

# New Prediction Logic

- Use exact matching result as a feature for score calculation.(1 or 0)
- Use exact matching **among synonyms** result as a feature for score calculation.(1 or 0)
- Use jaccard similarity result as a feature for score calculation.([0,1])

# New Prediction Logic

- IF calculated score is greater than threshold

  -return that candidate

- ELSE calculate **max** cosine similarity

# (1) Exact Matching

The same matching logic with the progress presentation is used.

- For each `'Habitat'` annotation in the *a1* files of the train data
- If annotated text exactly matches a term's name in the ontology (*obo* file)
- Then predicts that term as referent to the annotation
- Example:
  - In *a1*: `('T4', 'Habitat', 'gastric mucosa')`
  - In ontology: `'OBT:001792' -> 'gastric mucosa')`
  - Predicts `'OBT:001792'` as referent term

# (2) Exact Similarity Matching

Similar to exact matching, but among synonyms of an entity.

- For each `'Habitat'` annotation in the *a1* files of the train data
- If annotated text exactly matches one of the synonyms of this term (including EXACT and RELATED ones)
- Then predicts that term as referent to the annotation
- Example:
  - In *a1*: `('T4', 'Habitat', 'Neisser +')`
  - In ontology: `id: OBT:000473`
  - `name: Neisser-positive`
  - `synonym: "Neisser +" RELATED [TyDI:57816]`
  - `is_a: OBT:000174 ! Neisser stain phenotype`
  - Predicts `'OBT:000473'` as referent term

# (3) Weighted Similarity Matching

Combination of 3 features that we used.

- Exact match, exact similarity match, jaccard similarity
- Exact_match=0/1 Exact_match_synonyms=0/1 Jaccard_coef=[0-1]
- We assigned weights and calculated scores based on weights
- Hyper-parameter optimization

# Bidirectional Encoder Representations from Transformers

- BERT:Neural network-based technique for natural language processing pre-training
- huggingface/transformers library provides state-of-the-art general-purpose architectures for NLP with over thousands of pretrained models in 100+ languages
- BioBERT : A language representation model for biomedical domain

# Dictionaries

- ID_Vector_Dict_Name={}

  ID-vector dict for representations from id's_name

- ID_Vector_Dict_Training_Data={}

  ID-vector dict for representations from id’s training data

- ID_Vector_Dict_Ontology_Classes={}

  ID-vector dict for representations from id’s ontology super classes

- If we can not find candidate with weighted similarity>threshold :

  -Find id with max cosine similarity (with a minimum cosine similarity threshold)

  by using our dictionaries

  -Return id

# Results

Development Set

| | |
|---|---|
| Habitats | 0.4727 |
| Habitats (exact) | 0.3529 |
| Habitats (new in dev) | 0.4530 |
| Habitats (new in test) | - |
| Habitats (only unique form-normalization) | 0.3897 |

Test Set

| | |
|---|---|
| Habitats | 0.4500 |
| Habitats (exact) | 0.3147 |
| Habitats (new in dev) | 0.2727 |
| Habitats (new in test) | 0.1429 |
| Habitats (only unique form-normalization) | 0.4079 |

# Things to improve

- Syntactic dependency parser, getting headwords and using those relations

# Thanks for your attention

Şadi Uysal

Enes Özcan

Sercan Ersoy