Shared Task Evaluation

Learning Machine Learning

Nils Reiter

CRETA
CENTER FOR REFLECTED TEXT ANALYTICS

September 26-27, 2018

Hackatorial

Submit Results / Final testing

1. Open test.py
2. Make sure you have uncommented the classifier you want to use:

```
#train the classifier
#trainer = NBTrainer(tokens_with_features)
trainer = DTTrainer(tokens_with_features)
```

Hackatorial

Submit Results / Final testing

1. Open test.py
2. Make sure you have uncommented the classifier you want to use:

```
#train the classifier
#trainer = NBTrainer(tokens_with_features)
trainer = DTTrainer(tokens_with_features)
```

3. Run the script in your terminal
   - General Syntax

     `python test.py TRAIN_CORPUS TEST_CORPUS TEAM_NAME`
   - Example

     `python test.py ../data/Parzival_train.tsv MHD ExampleTeam`

Hackatorial

Submit Results / Final testing

1. Open test.py
2. Make sure you have uncommented the classifier you want to use:

```
#train the classifier
#trainer = NBTrainer(tokens_with_features)
trainer = DTTrainer(tokens_with_features)
```

3. Run the script in your terminal
   - General Syntax

     `python test.py TRAIN_CORPUS TEST_CORPUS TEAM_NAME`
   - Example

     `python test.py ../data/Parzival_train.tsv MHD ExampleTeam`

- Test corpora
  - Parzival: MHD
  - Werther: W
  - Bundestag: BTD

Section 2

Context/Applications/Entities

Preparations

- Annotation
  - Manually, following guidelines, with an annotation tool
  - Stored with character offsets

Preparations

- Annotation
  - Manually, following guidelines, with an annotation tool
  - Stored with character offsets
- Preprocessing
  - Sentence splitting, part of speech tagging, lemmatization
  - Done automatically, MHG tagger developed Echelmeyer et al. (2017)

Preparations

- Annotation
  - Manually, following guidelines, with an annotation tool
  - Stored with character offsets
- Preprocessing
  - Sentence splitting, part of speech tagging, lemmatization
  - Done automatically, MHG tagger developed Echelmeyer et al. (2017)
- Conversion into TSV format
  - Putting everything into a reasonable data structure
  - Preparing some information to be available directly
  - Let's look into the data structure

Preparations

- Annotation
  - Manually, following guidelines, with an annotation tool
  - Stored with character offsets
- Preprocessing
  - Sentence splitting, part of speech tagging, lemmatization
  - Done automatically, MHG tagger developed Echelmeyer et al. (2017)
- Conversion into TSV format
  - Putting everything into a reasonable data structure
  - Preparing some information to be available directly
  - Let's look into the data structure

TSV/CSV vs. CoNLL

- It's actually not a real table
  - Sentence boundaries are marked with an empty line
  - Most CSV/TSV libraries will skip those!
- Commonly used in NLP

Entity References → Entities

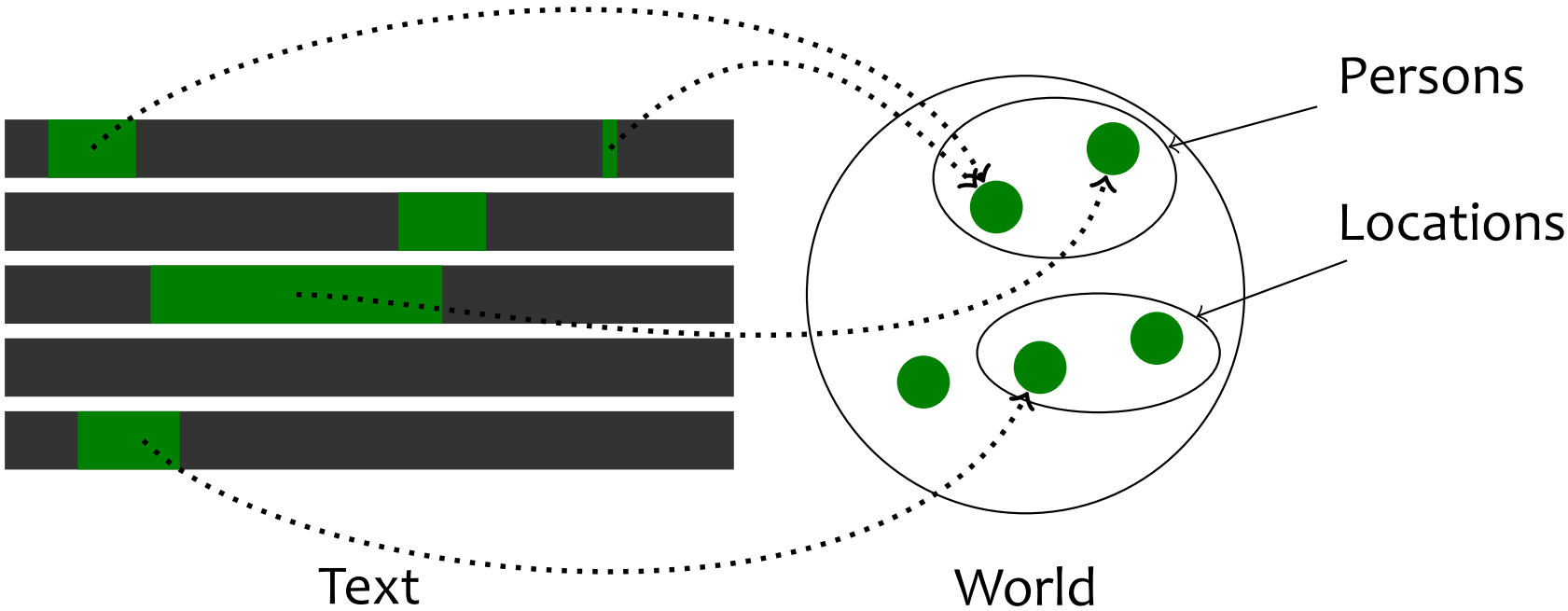

Figure: Entity references and entities

The Missing Link

si brâhte dar durch vlühtesal [des werden Gahmuretes kint]$_{PER}$.

The Missing Link

si brâhte dar durch vlühtesal [des werden Gahmuretes kint]$_{PER}$.

- Knowing that ‘des werden Gahmuretes kint’ refers to a person is nice, but not enough
- We want to know exactly which person it refers to (in the discourse context/the fictional world)

The Missing Link

si brâhte dar durch vlühtesal [des werden Gahmuretes kint]$_{PER}$.

- Knowing that 'des werden Gahmuretes kint' refers to a person is nice, but not enough
- We want to know exactly which person it refers to (in the discourse context/the fictional world)
- Can we make this a classification task?
  - What are the objects to classify?
  - What are the classes?
  - Where to we get features? What are potential features?

Option 1: Entities as classes

- Each discourse entity is a class
- Classification of entity references into these classes
- Features
  - Surface, lemma, context, meaning representation, gender, …

Option 1: Entities as classes

- Each discourse entity is a class
- Classification of entity references into these classes
- Features
  - Surface, lemma, context, meaning representation, gender, …
- Direct answer to the problem
- Difficult to generalize
  - Can only be trained/applied within the same discourse!
  - Nothing to be gained from one text to the next
    - Unless they have the same set of entities
- Established task in BioNLP: Entity tagging

Option 2: Detect Co-Reference

- Binary classification: True/False
- Classification of pairs of entity references into these classes
  - Do these two entity references refer to the same thing?
- Features
  - Case, number, gender, meaning, context, …

Option 2: Detect Co-Reference

- Binary classification: True/False
- Classification of pairs of entity references into these classes
  - Do these two entity references refer to the same thing?
- Features
  - Case, number, gender, meaning, context, …
- Generalization across texts
  - The computer can learn something about a general language phenomenon
- Needs post-processing: Naming the co-referent entities
- Established task in NLP: Coreference Resolution
  - Linguistically motivated, theory available

Modularization!

- NLP: Highly modularized pipelines
  - Each module can be/include a machine learning step
  - Specific, small tasks
- Dependencies between linguistic layers

References I

Echelmeyer, Nora, Nils Reiter, and Sarah Schulz. "Ein PoS–Tagger für „das" Mittelhochdeutsche". In: *Book of Abstracts of DHd 2017*. Bern, Switzerland, Feb. 2017.