



Emanuele Alessi

NLP HW3

Presentation

Project tasks

- ▶ Mandatory task (role labeling)
- ▶ Extension 2
- ▶ Create an alignment between BabelNet synsets and PropBank predicates
- ▶ Predicate identification and disambiguation

A thick orange diagonal stripe runs from the top right corner towards the bottom left, separating the white background on the left from the solid orange background on the right.

1.

Mandatory task

Neural architecture

- ▶ Embeddings layer (Glove + POS tags + predicate flags)
- ▶ BiLSTM
- ▶ Softmax classifier

Algorithm

1. Parse CoNLL 2009 dataset
2. Convert lemmas, POS tags and roles into integer values
3. Save them into a list
4. Divide the list into k sub-lists
5. Train the BiLSTM passing lemmas + POS tags + predicate flags (batches) and roles (labels)
6. Save the model

Results

Batch size	Hidden size	Optimizer	Learning rate
10	128	Adam	0.001

F1 (Macro)	Precision	Recall	Accuracy
76,8 %	67,7 %	88,8 %	96,6 %



2.

Extension 2

HW2 system

- ▶ Embeddings layer (Glove + POS tags)
- ▶ BiLSTM
- ▶ Softmax classifier

Hyper parameters

Batch size	Hidden size	Optimizer	Learning rate
10	100	Adam	0.001

Results

	F1	Accuracy
Senseval2	69,2 %	81,4 %
Senseval3	70,5 %	84,8 %
Semeval2007	66,2 %	92,8%
Semeval2013	63,0 %	89,4 %
Semeval2015	64,6 %	79,6 %

BabelNet to PropBank alignment

```
wsd = HW2_instance()
```

```
d = {}
```

```
for each sentence in CoNLL 2009 do
```

```
    predictions = wsd.predict(lemmas of the sentence)
```

```
    for i = 1 to predictions length do
```

```
        if predictions[i] ≠ 'UNK' and predicate of the sentence ≠ '_' then
```

```
            d[predictions[i]].append(predicate of sentence[i])
```

```
for each pair (synset, predicates) in d do
```

```
    write a line in babelnet2propbank.txt which contains the synset and the  
most common predicate in predicates
```

A large, solid pink shape that starts as a thin diagonal line from the top left and expands into a wide triangular area on the right side of the slide.

3.

Predicate classification

Neural architecture

- ▶ Embeddings layer (Glove + POS tags)
- ▶ BiLSTM
- ▶ Softmax classifier

Two different approaches

- ▶ Train the predicate classifier using batches and labels in CoNLL + SemCor
- ▶ Train the predicate classifier using batches and labels which are contained ONLY in CoNLL 2009

Results

Batch size	Hidden size	Optimizer	Learning rate
10	100	Adam	0.001

F1 (Macro)	Precision	Recall	Accuracy
88,6 %	85,3 %	92,1 %	95,7 %



Thanks for your attention