Shallow from Deep

Liling Tan
Universitaet des Saarland

ACE + ERG Lemmatizer

How?

- Using RELS list from ACE output and ERG PRED-ORTH mappings to get lemmas

Why?

- Using DELPH-IN tools for shallow NLP
- ACE is easy to use

```
>>> sent = 'the geese ate a ratatta with two bigger
bottles of wine that are larger than life and the
sentence went on and on.‘

>>> ace_lemmatize(sent,onlylemma=True)
['the', 'goose', 'eat', 'a', 'ratatta', 'with', 'two', 'bigger',
'bottle', 'of', 'wine', 'that', 'are', 'larger', 'than', 'life',
'and', 'the', 'sentence', 'go', 'on', 'and', 'on']
```

ACE + ERG WSD

How?

- Using KEYREL.PRED values from ACE output to get chunks as vectors for Lesk algorithm

How good?

- With some hacks, from my straw evaluation, it works the same as **`simple_lesk()`**

Work in Progress

- ACE + ERG Chunker
  - Using ARG relations from ACE output to get chunks
- ACE + GG Lemmatizer / Chunker
- Other ACER-G NLP applications?

Summary

- Using DELPH-IN tool for NLP tasks
- Deep HPSG parsing gives good shallow analysis

Fin.

Shallow Tools, Deep Grammar

- Other than parsing, how can DELPH-IN tools interact with the wider NLP task/application driven world?

- Using supertags/uebertags as feature sets for classification based NLP task?

- Creating a super flattened / simplified token-annotation based parse output

- Reviving LOGON, encouraging more DELPH-IN based MT system?

- Using MRS as feature sets for semantic related task?
  - WSD, Entity linking, Knowledge Base Population with DELPH-IN?