Analyzing German Noun Compounds using a Web-Scale Dataset – Task description

UIMA Software Project WS 2010/2011

Introduction

- Noun-compounding: Combination of two existing words to another new word.
- Powerful feature in the German language
- Example: *Blumensträuße* (flower bouquet) -> *Blumen* (flower) + *Sträuße* (bouquet)

Problem in many NLP task

- Search for a compound word should also include result with the words splitted
- Example: *Lackschicht* (paint layer) should return results with the words *Lackschicht* and *Schicht aus Lack* (layer of paint)

Problem definition

- Compounds are formed with nouns, verbs and adjectives.
- Compound words can be compound with other
- Linking morphemes are added between words: *Tag(es)+ration*
- Different context for different splits: *Tag(es)+ration* vs. *Tag(es)+rat+ion*

Main algorithm [ea08]

1. Calculate every possible way of splitting a word in one or more parts
2. Score those parts according to some weighting function
3. Take the highest-scoring decomposition. If it contains one part, it means that the word in not a compound.

Roadmap

Week	Goals
08.11 - 14.11.	get familiar with the project; choosing dictionary
15.11 - 21.11.	access to dictionary
22.11 - 28.11.	access Google Web1T; splitting words
29.11 - 05.06.	splitting words
06.12 - 12.12.	splitting words
13.12 - 19.12.	evaluation and testing
20.12 - 26.12.	weighting function (Christmas)
27.12 - 02.01.	weighting function (Christmas, new years eve)
03.01 - 09.01.	weighting function
10.01 - 16.01.	no time (vacation)
17.01 - 23.01.	evaluation and testing
24.01 - 30.01.	UIMA Component
31.01 - 06.02.	project cleanup

End

Questions

Ask now, or later.

More information

Code, documentation and slides are available on github:
`https://github.com/jenshaase/noun-decompounds`

References

Enrique Alfonseca et al.
German decompounding in a difficult corpus.
In *Computational Linguistics and Intelligent Text Processing*, 2008.