Analyzing German Noun Compounds using a Web-Scale Dataset – Report Week 2

TECHNISCHE UNIVERSITÄT DARMSTADT

UIMA Software Project WS 2010/2011
Jens Haase

1 Work done in the last week

1.1 Corpus and Evaluation

Before I want to start to create a first splitting algorithm, I want to have a method to evaluate the algorithm. For that the corpus of Marek `http://diotavelli.net/files/ccorpus.txt` is used. A reader reads the file and creates a object structure for each split. The word annotation like, {N} or {A,N} are removed. For evaluation we are only interested in the splits and the linking morphemes, and not the word form.

With the corpus it is now possible to create a evaluation class for the splitting algorithm. A splitting algorithm always returns a list of all splits. The evaluation class will check if the correct split is in the list of all possible splits.

1.2 Simple splitting algorithm

Starting with a simple splitting algorithm has the benefits that a lot of possible splitting cases can be found. The algorithm than can be improved from simple to advanced.

The first draft of my algorithm iterates from left to right over the word. It splits the word at the current position and checks if the left word is in a dictionary. If the left word is in the dictionary the right side will be evaluated with the same algorithm. If the right side can not be evaluated we check if the beginning of the right word contains a morpheme. The morpheme then will be removed from the right word and the smaller right word will be evaluated again. At the end we have a split with the smallest word forms.

In the next step we try to recombine neighbors of the split. *Aktionsplan* will be evaluated by the first part to *Akt+ion(s)+plan*. The recombine will now combine *Akt* and *ion(s)* to *Aktion(s)* and check if the word is in the dictionary. If true we add this new split to the list of all possible splits.

At the end the algorithm returns a list of all possible splits.

1.3 Current Evaluation

The current evaluation of the splitting algorithm has a recall of 0.45. This is currently not very much, but there are a few bugs in the recombination of words. Another big problem is that not all needed words are in the dictionary. That means that most of the words will never be splitted. For example, the dictionary contains *berechnen* but not *berechnung* and because of that *berechnungsarten* can not be splitted. The word *berechnung* can be calculated in using the flags behind each word in the IGerman98 Dictionary. This requires some extra work, which is not done yet.

The good thing is that I only found one totally false splitted word in the first 200 words. The word *minimalanforderungen* must be splitted to *minimal+anforderung(en)* but is currently splitted to *minima+lan+ford(e)+run+gen*. The recombination will here also not help.

1.4 Time Tracking

Date	Task	Planned time
2010-11-18	Evaluation Corpus, Splitter Evaluation	4
2010-11-20	Simple Splitter	4
2010-11-21	Simple Splitter	4
2010-11-22	Evaluating current status of splitting algorithm	1
2010-11-24	Report writing	2

2 Plan for next week

As mentioned above the current dictionary is not optimal. The IGerman98 Dictionary can be improved by using the flags. Another option will be the Google Web1T unigrams.

With a better dictionary the recall of the splitter will hopefully be better. The we can decide if the splitter can be improved or we have to use a total different splitting algorithm. Other splitting options are:

- Do the same from right to left
- Combine result from “Left to right” and “right to left”
- Use a data-driven algorithm as mention by Larson et al. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.7240\&rep=rep1\&type=pdf`

Date	Task	Planned time
2010-11-25	Better dictionary reader	4
2010-11-26	Improve Splitter	4
2010-11-27	Improve Splitter or try complete other algorithm	4
2010-11-29	Evaluating current status of splitting algorithm	1
2010-11-31	Report writing	2