Analyzing German Noun Compounds using a Web-Scale Dataset – Report Week 9

TECHNISCHE UNIVERSITÄT DARMSTADT

UIMA Software Project WS 2010/2011
Jens Haase

1 Work done in the last week

1.1 Ranking algorithm on tree

Currently the ranking algorithm works only on lists. But the splitting algorithm returns also a tree. With a tree based ranking algorithm we can reduce the amount of calculation, when we walk along the tree.

The current implementation takes the parent and all children and ranks these items. If the item with the highest rank is equal to the parent, we can return the parent as the total best item. If one of the children has the highest rank we recursively call the tree ranking function with the child as new parent.

This method walks down the tree on one path until a parent is higher ranked as it's children or it reaches a leaf. The disadvantages of this method is that not all possibilities will be evaluated. The following tables shows the results:

Algorithm	**Correct tree**	**Correct list**
Frequency Based	0.53 (0.732)	0.511 (0.698)
Probability Based	0.173 (0.238)	0.175 (0.243)
Mutual Information Based	0.358 (0.521)	0.419 (0.603)

The difference between the list and the tree method are nearly equal. In the frequency based algorithm the tree method is a little bit better, while the mutual information based algorithm is a little bit worse.

1.2 Combine ranking algorithm

As next step I liked to combine all ranking algorithms. After reading some papers I came to the solution that only a machine learning algorithm can solve this problem. A learning algorithm can be trained to split a word in two parts with a bi-label classifier.

This requires a lot of extra work because the current evaluation dataset only gives the complete split of a word. Because the final delivery is near I decide to skipped this task.

1.3 Testcases and refactoring

At least I did some refactoring and added some testcases. All testcases that require external data or tools are ignored if they are not found. This can decrease the test coverage but do not result in errors. With all tools and data installed the current test case coverage is 80.4 % (line based).

1.4 Time Tracking

Date	**Task**	**Needed time**	**Planned time**
2011-01-20	Tree based method	4	6
2011-01-21	Combine algorithm	5	3
2011-01-26	Testing and refactoring	3	0
2011-01-26	Report Writing	2	3

2 Plan for next week

In the following week I have to implement a UIMA component with the current algorithms. Also further code cleanup and JAVA documentation is planned.

Date	Task	Planned time
2011-01-27	UIMA Component	4
2011-01-28	Refactoring	4
2011-01-29	Code cleanup	4
2011-01-30	Java doc	2
2011-01-02	Report Writing and final delivery	3