Full Tagging of Real Data Using Memory-Based Learning

Jiří Mírovský, Charles University in Prague, Czech Republic

mirovsky@ufal.mff.cuni.cz

Tagging

Tagging is a principal task and a basic step in most approaches to NLP. One of the most successful solutions of this task, HMM tagger, makes morphological analysis first and then selects the right tag from its ambiguous output using n-gram statistical model and Viterbi algorithm [1]. We tried to achieve the same goal in a slightly different way, using Memory-Based Learning.

Memory-Based Learning

Memory-Based Learning (MBL) is a lazy method of machine learning. It means that during the training phase it only stores training examples into memory. During testing, new examples are classified by extrapolation from the most similar stored cases. The definition of similarity is crucial to the success.

Solution

In MBL, each example is represented by a set of attributes.
In our case, the attributes are important parts of words close to the word we want to assign a tag to and of course the word (or its part) itself.
We experimented with different number of attributes and with different length of the parts of words. The parts of words were always formed from the fixed-length suffixes of the words.
The right tag was chosen from *k-nearest neighbours* (second column in the result tables) with *inverse distance weighting*. The *modified value difference metric* was used for measuring difference between values of attributes. For attributes we used the *information gain weighting*.

TiMBL

Tilburg Memory-Based Learner is a very good tool for experimenting with MBL.
It was developed by [2].
TiMBL implements many MBL techniques and algorithms and allows users to set many parameters.

Formulas

$$\Delta(X,Y) = \sum_{i=1}^{n} w_i \, \delta(x_i, y_i)$$

Distance between two examples (weighted sum over attributes)

$$\delta(v_1, v_2) = \sum_{i=1}^{n} |P(C_i|v_1) - P(C_i|v_2)|$$

Modified Value Difference Metric (distance between values)

Results

We used three different data sets. Two small sets (in English and in Czech) and one large set in Czech.
The results are displayed in three tables, one table for each data set. The numbers in the first column mean the length of suffixes of the particular words, each number stands for one word. For example, 4-5-4 means that one training/test example consisted of suffixes of three successive words of given lengths. The underlined number marks the word whose tag is in consideration. With this setting, the sentence "`The procedure causes great uncertainty.`" forms six traning examples (one for each word and the full stop); one of them is: "`dure auses reat VBZ`", the next one is: "`uses great inty JJ`", etc.

 # train examples: 237 478
 # test examples: 10 000
train language: English

attributes	k	precision (%)	exact matches	examples/s
2-3-4-4	20	92.2	18	5.6
2-4-5-4	15	93.1	17	4.7
2-4-5-4	20	93	17	4.6
2-4-6-4	1	92.5	18	7.4
2-4-6-4	3	93.1	18	5.4
2-4-6-4	5	93	18	5.3
2-4-6-4	10	93.2	18	5.3
2-4-6-4	15	93.2	18	5.1
2-4-6-4	20	93.2	18	4.5
2-4-6-4	30	93.1	18	4.6
2-4-7-4	10	93	18	4.3
2-4-7-4	15	93	18	4.2

 # train examples: 374 311
 # test examples: 10 000
train language: Czech

attributes	k	precision (%)	exact matches	examples/s
3-4	1	82.8	72	37.5
3-3-5	15	84.1	30	1.7
3-4-3	15	86.5	28	3.8
4-5-3	10	86.5	27	3.8
2-3-4-4	5	86.8	10	1.1
2-3-4-4	15	86.9	10	0.9
2-3-4-4-3	5	87	2	0.6
2-3-4-4-3	10	87.2	2	0.5
4-4-4-4-4	15	87.2	2	1.4
5-5-5-5-5	5	87	2	2
10-10-10-10	5	83.5	6	1.1
4-4-4-4-4-4-4	10	87.1	0.3	0.5

 # train examples: 1 564 780
 # test examples: 137 734
train language: Czech (PDT 1.0 standard train & devtest)

attributes	k	precision (%)	exact matches	examples/s
2-3-4-4	5	88.6	22412	0.16
3-4-4-4-3	5	89.4	5883	1.31
3-4-4-4-3	10	89.6	5883	1.1
4-4-4-4-4	1	88.46	5475	2.68
4-4-4-4-4	5	89.56	5475	1.7
4-4-4-4-4	10	89.67	5475	1.38
4-5-6-5-4	5	89.58	5633	0.97
4-5-6-5-4	10	89.76	5633	0.66
4-5-6-5-4	15	89.77	5633	0.6
5-6-7-6-5	5	88.89	5493	0.67
5-6-7-6-5	10	89.1	5493	0.44
5-6-7-6-5	20	89.08	5493	0.6

Conclusion

As expected, Czech has again proved to be significantly more difficult for tagging than English.
The results on the two smaller data sets were comparable to results obtained by HMM tagger (our work).
On larger Czech data, the results of HMM tagger [1] are about 5% better than our results, but there is still a room for an improvement, as suggested in Future Work.
TiMBL has prooved to be a powerfull and stable tool for basic experiments with MBL, though some important features are missing – see again Future Work.

Future Work

The most promissing way how to improve the results is to include the morphological analysis so that TiMBL could only choose from tags offered by the morphological analyzer. Unfortenately, TiMBL itself has no feature which would support it and it has to be done indirectly.
There is one other feature lacked in TiMBL which (if it existed) would be able to help improve the results: It would be great if the output from the previously evaluated test examples (tag) could be used as a part of the actual test example.

References

[1] Jan Hajič, Pavel Krbec, Karel Oliva, Pavel Květoň, Vladimír Petkevič (2001): Serial Combination of Rules and Statistics: A Case Study in Czech Tagging. In Proceedings of ACL 2001.

[2] Walter Daelemans, Jakub Zavřel, Ko van der Sloot, and Antal van den Bosch (2004): TiMBL: Tilburg Memory Based Learner, version 5.1, Reference Guide. ILK Technical Report 04-02.

[3] Jan Hajič, Barbora Vidová-Hladká, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas (2001): Prague Dependency Treebank 1.0 (Final Production Label). In CDROM CAT: LDC2001T10.