1 Baseline

The baseline proposed by the organizers of the CoNLL 2000 shared task got pretty high score overall. But they do only apply simple methods and no ml methods with ex. deep neural networks. (Which is understandable since the conference was held year 2000.)

Our own simple baseline evaluation gave 0.7729 in accuracy and FB1 score of 77.07. Applying conlleval script gave following output:

```
perl conlleval.txt <out
processed 47377 tokens with 23852 phrases; found: 26992 phrases; correct: 19592.
accuracy:  77.29%; precision:  72.58%; recall:  82.14%; FB1:  77.07
             ADJP: precision:   0.00%; recall:   0.00%; FB1:   0.00  0
             ADVP: precision:  44.33%; recall:  77.71%; FB1:  56.46  1518
            CONJP: precision:   0.00%; recall:   0.00%; FB1:   0.00  0
             INTJ: precision:  50.00%; recall:  50.00%; FB1:  50.00  2
              LST: precision:   0.00%; recall:   0.00%; FB1:   0.00  0
               NP: precision:  79.87%; recall:  86.80%; FB1:  83.19  13500
               PP: precision:  74.73%; recall:  97.07%; FB1:  84.45  6249
              PRT: precision:  75.00%; recall:   8.49%; FB1:  15.25  12
             SBAR: precision:   0.00%; recall:   0.00%; FB1:   0.00  0
               VP: precision:  60.53%; recall:  74.22%; FB1:  66.68  5711
```

Figure 1: Outprint of command: *perl conlleval.txt < out*

2 Using Machine Learning

The feature vector in ml_chunker is similar but not exactly the same as Kudoh and Matsumoto, 2000. Ml_chunker are missing features that Kudoh and Matsumto are using, the values of the two previous chunk tags in the first part of the window: c i-2 , c i-1.

The accuracy using logistic regression is 91% (compared with 77.29% when we used baseline method, pretty good improvement) and FB1 score 77.07.

Removing the lexical features (the words) from the feature vector results in an accuracy of 90% and FB1 score 75.97.

We tried implement decision trees and perception model resulting in accuracy 94.79% and FB1 score 91.05 respectively 93.62% for decision tree model and FB1 score 88.85 for perception model.

3 Improving the Chunker

We chose to complement the feature vector used in the previous section with the two dynamic features, c_{i-2} , c_{i-1}. Here we needed to extract an new feature vector after every prediction as it was dependent on previous predictions (c_{i-2} , c_{i-1}). Using logistic regression gave result below, evalutated with conlleval.txt.

```
Predicting the test set...
perl conlleval.txt <outml_improved
processed 47377 tokens with 23852 phrases; found: 23882 phrases; correct: 22037.
accuracy:  95.02%; precision:  92.27%; recall:  92.39%; FB1:  92.33
             ADJP: precision:  79.27%; recall:  64.61%; FB1:  71.19  357
             ADVP: precision:  80.09%; recall:  78.98%; FB1:  79.53  854
            CONJP: precision:  71.43%; recall:  55.56%; FB1:  62.50  7
             INTJ: precision: 100.00%; recall:  50.00%; FB1:  66.67  1
              LST: precision:   0.00%; recall:   0.00%; FB1:   0.00  0
               NP: precision:  92.21%; recall:  92.75%; FB1:  92.48  12494
               PP: precision:  96.14%; recall:  97.73%; FB1:  96.93  4891
              PRT: precision:  80.00%; recall:  75.47%; FB1:  77.67  100
             SBAR: precision:  88.93%; recall:  84.11%; FB1:  86.46  506
               VP: precision:  92.27%; recall:  92.55%; FB1:  92.41  4672
```

Figure 2: Outprint of perl conlleval.txt < outml_improved

4 Reading

There is difference from the given paper, Contextual String Embeddings for Sequence Labeling by Akbik et al. (2018), and our method. Firstly, the authors used a recurrent neural network, instead of logistic regression as us. Secondly their method use the embeddings with distinct properties that *"(a) are trained without any explicit notion of words and thus fundamentally model words as sequences of characters, and (b) are contextualized by their surrounding text, meaning that the same word will have different embeddings depending on its contextual use."*. This method compared to ours, where we trained our model with explicit notion of words, trained with explicit notion of characters. With this method they reached a FB1 score of 96.72 ± 0.05.