We describe our approach taken in the MSM2013 IE Challenge, which was aimed at concept extraction from microposts - a short in size information posted on the Web (e.g. Tweets, Facebook likes, comments, Google+ posts, Instagram photos). The classification of extracted concepts has been restricted to four classes: PER, LOC, ORG and MISC.

**Approach**

Our intent was not to create a new NER tagger, but to combine several existing NER taggers which we evaluated over the MSM 2013 training set. We have tried several algorithms to train a classification model:

- MultiLayer Perceptron Neural Network
- C4.5
- SimpleCart
- Random Forest
- LADTree (Logistic Model Trees)
- NBTree (Bayesian Network Tree)
- REPTree (Fast decision tree learner)
- …

Each of these approaches has its own pros and cons, so we have tried to combine them in such a way that they still respect each other’s classification methods and benefit from their combination. We felt, that this approach could bring a superior performance. This idea was supported by evaluation results of various NER taggers, which we evaluated over the MSM 2013 training set.

The evaluation has showed, that total recall of all the taggers combined together was much higher than individually. This ment that taggers discovered diverse entities and that there could be a place for combining the taggers in order to achieve superior performance. The problem was to improve the precision, which was very low. Better performance could be easily achieved by a straightforward combination of the best taggers for each NE type, but the performance growth would not be significant. Therefore we tried to combine the taggers in more sophisticated way through machine learning techniques.

**Features for ML**

The point is to describe a) micropost text as a whole; b) annotations found by involved NER taggers; c) how these annotations overlap with each other. Annotations extracted by NER taggers: LOC, MISC, ORG, PER, NP, VP, OTHER. We use an d) answer vector for learning.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1A</td>
<td>Recall of all the taggers combined together</td>
</tr>
<tr>
<td>F1L</td>
<td>Recall of all the taggers combined together</td>
</tr>
<tr>
<td>F1S</td>
<td>Recall of all the taggers combined together</td>
</tr>
<tr>
<td>RL</td>
<td>Recall of all the taggers combined together</td>
</tr>
<tr>
<td>PA</td>
<td>Precision of all the taggers combined together</td>
</tr>
</tbody>
</table>

**Model training**

- Input data: 36,000 instances
- Preprocessing of input data: removed duplicate instances
- Preprocessed input data, ready for ML: 200 attributes
- Validation: 10-fold cross validation holdout

The best models were built by: Random Forest, C4.5

The graph "Micro summary" and the table below compare performance of NER taggers over the MSM 2013 test set. This set was modified prior to evaluation. There were duplicate tweets and tweets overlapping with the train set removed.

Annotowatch 1, 2 and 3 are our submissions to the MSM 2013 Challenge, which used post processing techniques described in the submission paper.

**Outlines of chosen NER taggers**

- **ANNE**
- **OpenNLP**
- **Stanford NER**
- **Wikipedia Miner**
- **Illinois NER**
- **Apache OpenNLP**
- **Open Calais**

**Transformation to machine learning task**

We have:

- unique 2752 manually annotated tweets
- NER taggers

We want:

- classification of the taggers' results to 5 classes: LOC, MISC, ORG, PER, NULL

- Combined NER taggers