MSM 2013 IE Challenge: Annotowatch

Stefan Dlugolinsky, Peter Krammer, Marek Ciglan and Michal Laclavik

Institute of Informatics, Slovak Academy of Sciences, Dubravska cesta 9, 845 07 Bratislava, Slovak Republic

WWW 2013 Rio de Janeiro, Brazil
22nd International World Wide Web Conference
13th -17th, May

Institute of Informatics Slovak Academy of Sciences

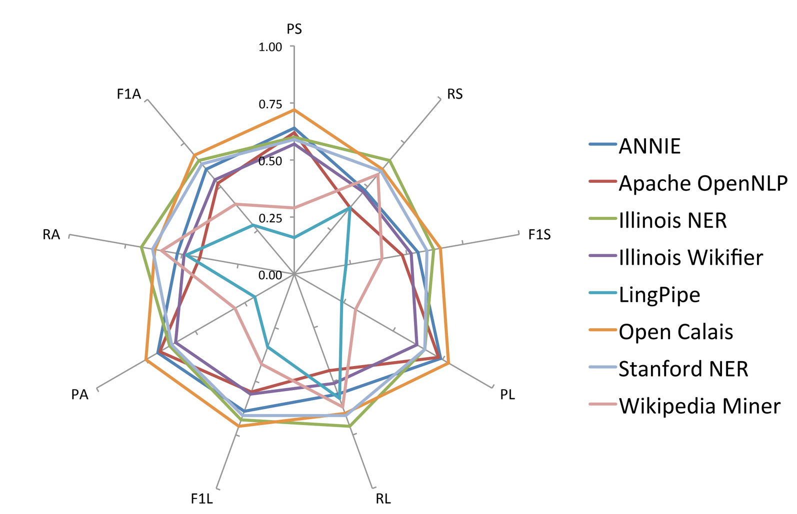
Introduction

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.

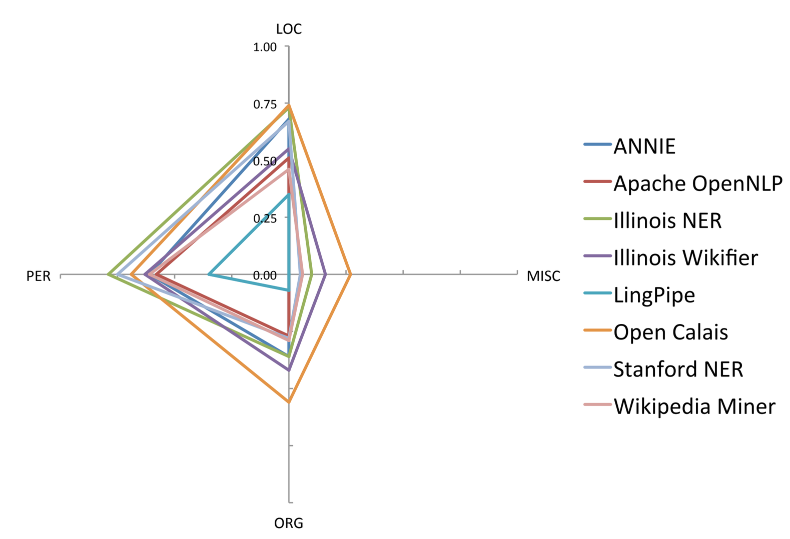
Approach

Our intent was not to create a new NER tagger, but to combine several existing NER taggers which use different 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.

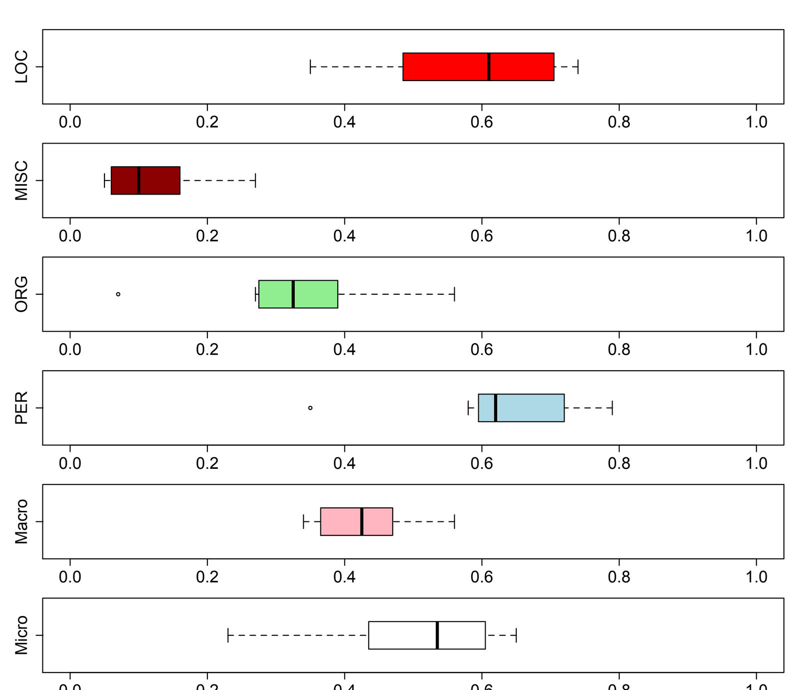
Evaluation of considered NER taggers for combination



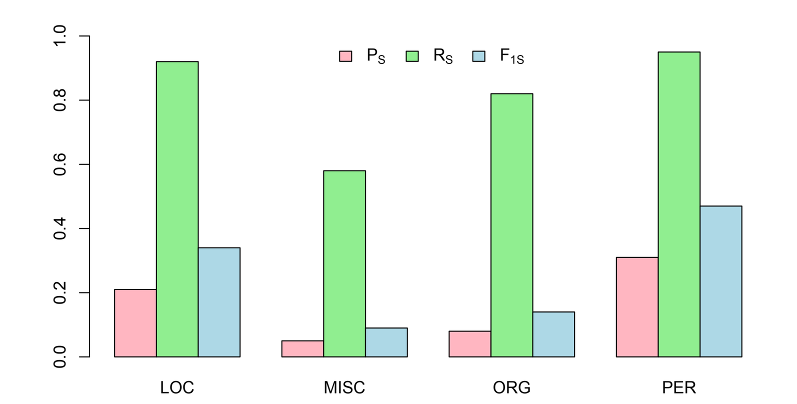
Micro summary of several NER taggers over the training set. Three versions of Precision, Recall and F_1 metrics are depicted. Strict (P_S, R_S and F_{1S}) considered partially correct responses as incorrect, lenient (P_L, R_L and F_{1L}) considered them as correct and P_A, R_A, and F_{1A} is average of previous two.



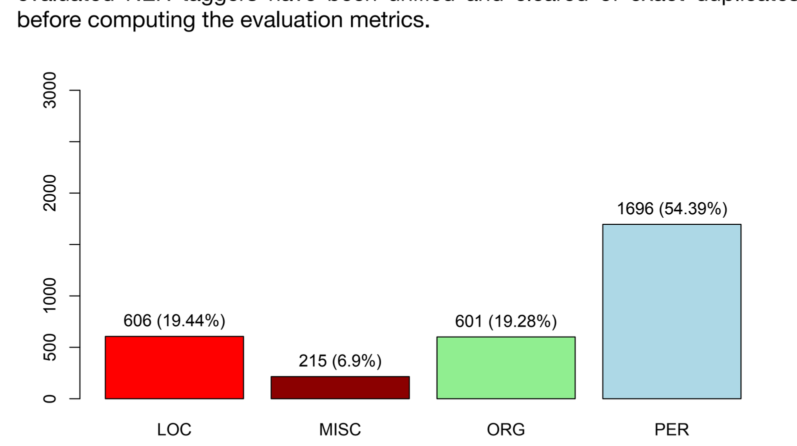
F1 score of the evaluated NER taggers over the training set.



F_1 spread of the evaluated NER taggers over the training set.



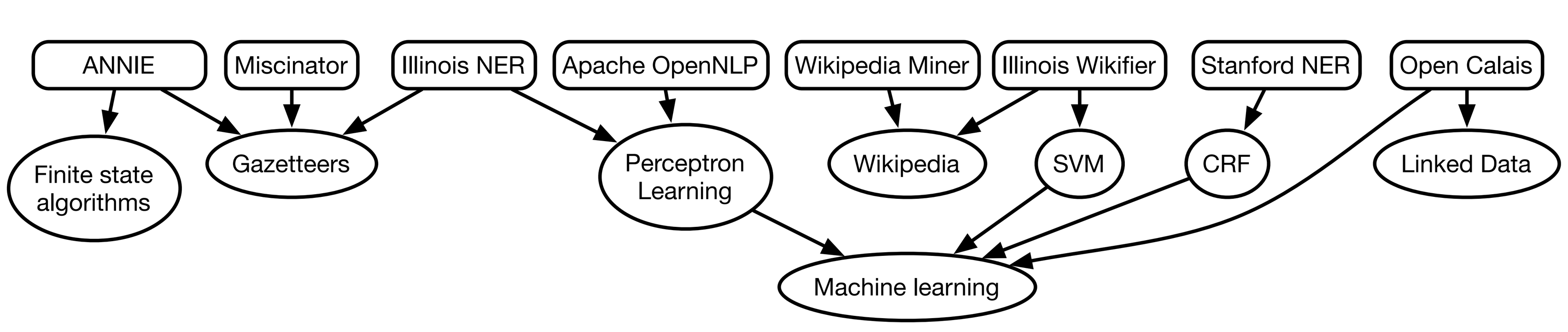
Performance of unified NER taggers over the training set. Results of all the evaluated NER taggers have been unified and cleared of exact duplicates before computing the evaluation metrics.



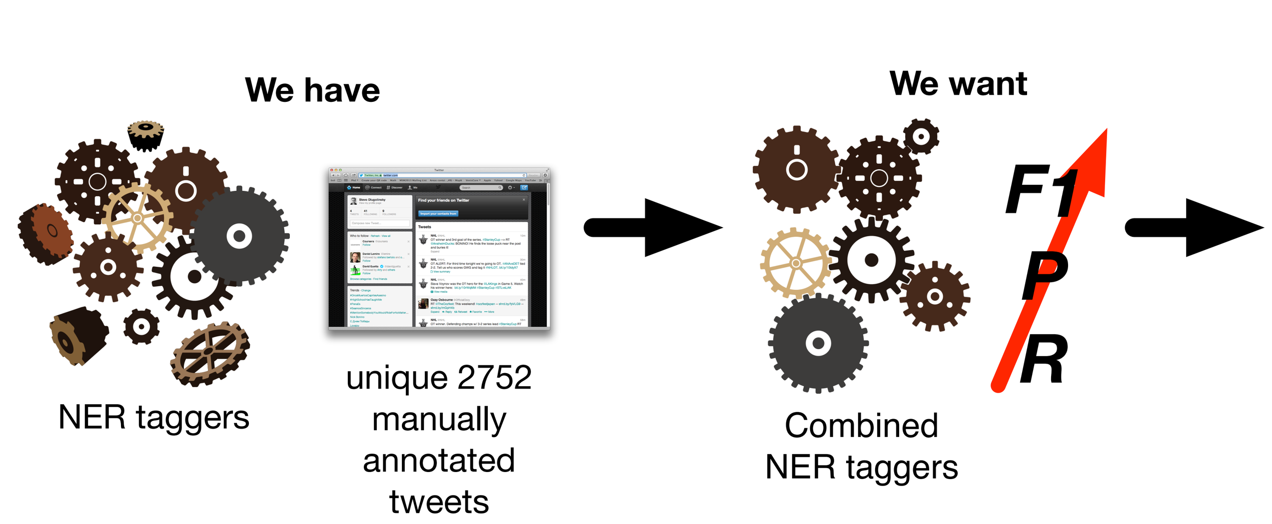
Occurrence of particular NE types in the training set.

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.

Outline of chosen NER taggers



Transformation to machine learning task



We do

Classification of the taggers' results to 5 classes: **LOC, MISC, ORG, PER, NULL**

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.

a) Tweet feature vector - Three features are computed considering only words with length > 3:

- **awc** - all words in tweet are capitalized
- **awuc** - all words in tweet are upper cased
- **awlc** - all words in tweet lower cased

b) Annotation feature vector - Six attributes computed for each annotation found by underlaying NER tagger (reference annotation):

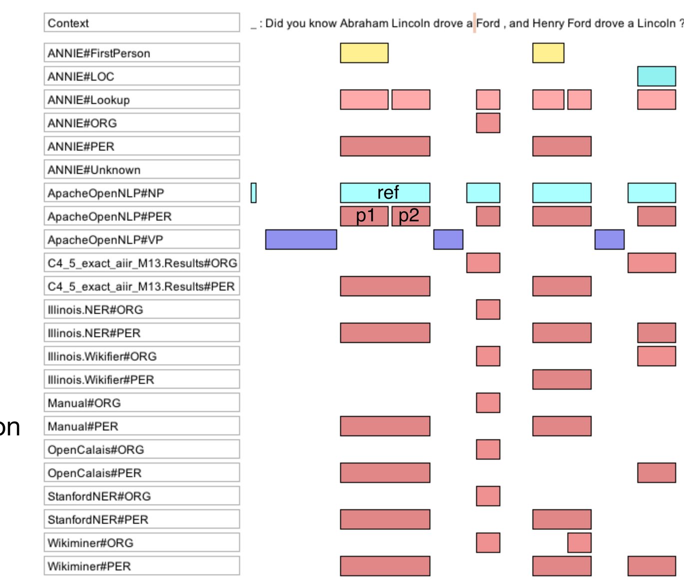
- **ne** - annotation class (LOC, MISC, ORG, PER, VP, NP, OTHER)
- **flc** - first letter capital
- **aluc** - all letters upper cased
- **allc** - all letters lower cased
- **cw** - capitalized words
- **wc** - word count

c) Tagger vector - Three attributes describing overlapping of reference annotation with other annotations and two attributes describing particular tagger confidence (per tagger and each NE class):

- **ail** - average intersection length
- **aiia** - average percentage intersection of other's annotations with reference annotation
- **aiir** - average percentage intersection of a reference annotation with other's annotations (100% is the length of the ref. ann.)
- **E(p)** - mean value of tagger confidence for overlapping annotations
- **var(p)** - variance of tagger confidence values for overlapping annotations

d) Answer vector - An intermediate help vector describing, how reference annotation overlaps with the manual annotations. This vector is used in ML data preprocessing and is later replaced by value of **ne** attribute or NULL:

- **ne** - NE class of manual annotation overlapped by the reference annotation
- **ail** - average intersection length with manual annotations
- **aiia** - average percentage intersection of manual annotations with reference annotation
- **aiir** - average percentage intersection of a reference annotation with manual annotations



Example of attribute computation in tagger vector for Apache OpenNLP and PER NE type. The reference annotation "ref" is of a NP type:

$$aiir = \frac{\frac{|ref \cap p_1|}{|ref|} + \frac{|ref \cap p_2|}{|ref|}}{2}$$

$$aiia = \frac{\frac{|ref \cap p_1|}{|p_1|} + \frac{|ref \cap p_2|}{|p_2|}}{2}$$

Model training

We have tried several algorithms to train a classification model:

C4.5
LMT (Logistic Model Trees)*
NBTree (Bayess Network Tree)
REPTree (Fast decision tree learner)
SimpleCart
LADTree (LogitBoost Alternating Decision Tree)
Random Forest
AdaBoostM1
MultiLayer Perceptron Neural Network
Bayes Network
Bagging Tree
FT (Functional trees)*

* trees supporting rules made of multiple attributes

Input data
~36,000 instances
200 attributes

Preprocessing of input data

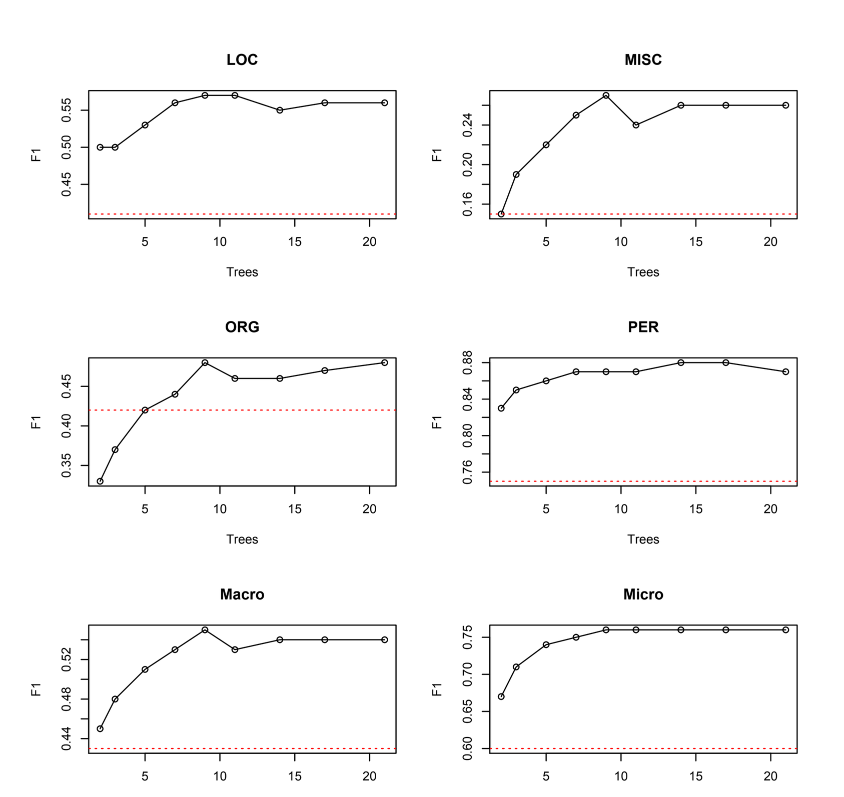
- removed duplicate instances
- removed attributes where values changed for insignificant number of instances
- removed help attributes
- attribute conversion to Nominal

Preprocessed input data, ready for ML
~31,000 instances
100 attributes

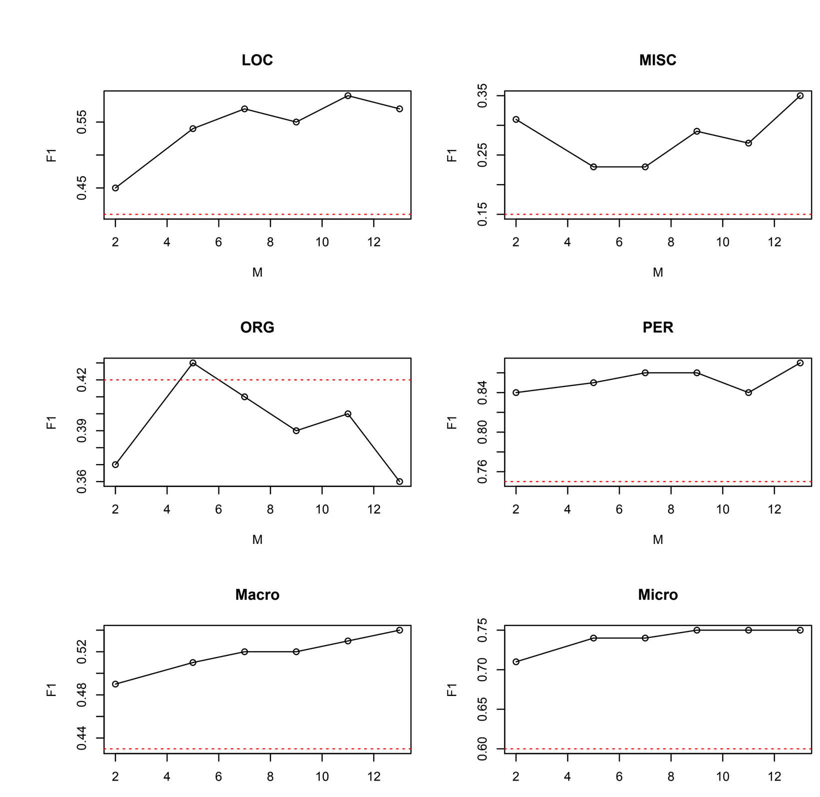
Validation
10-fold cross validation
holdout

The best models were built by:

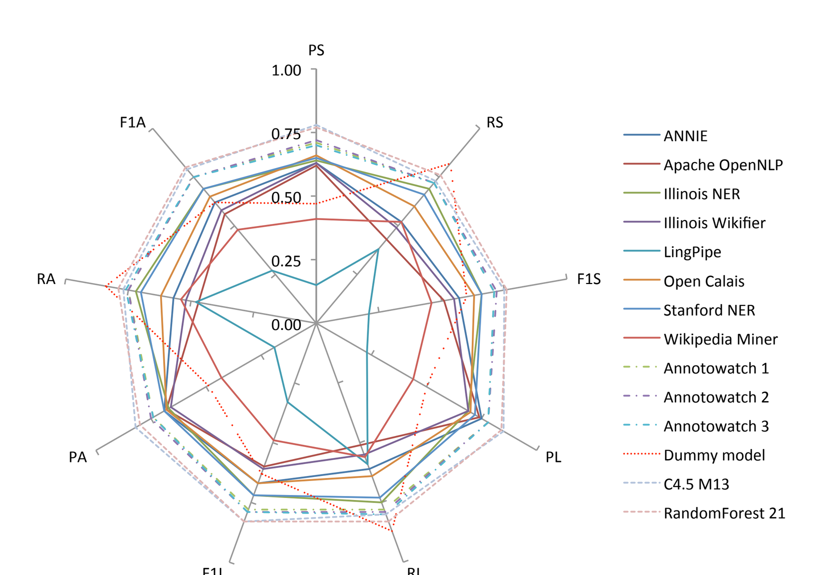
Random Forest



C4.5



Trees - number of trees; M - minimum number of records to be passed to child decision nodes; --- Dummy model; Trained on MSM train dataset, tested on MSM test dataset



Micro summary

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.

RandomForest 21 and C4.5 M13 are our new models created after the challenge submission deadline. They do not involve any post processing of the results.

"Dummy" is a tagger which simply combines best taggers for each NE class (based on evaluation over the train set).

Tagger	LOC			MISC			ORG			PER			Macro			Micro		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
RandomForest 21	0.51	0.61	0.56	0.39	0.19	0.26	0.50	0.47	0.48	0.86	0.88	0.87	0.56	0.54	0.54	0.77	0.76	0.76
C4.5 M13	0.53	0.61	0.57	0.59	0.25	0.35	0.41	0.33	0.36	0.87	0.87	0.87	0.60	0.51	0.54	0.78	0.73	0.75
Annotowatch 2	0.39	0.54	0.46	0.38	0.25	0.30	0.39	0.40	0.40	0.85	0.85	0.85	0.50	0.51	0.50	0.72	0.72	0.72
Annotowatch 1	0.44	0.58	0.50	0.39	0.26	0.31	0.39	0.40	0.39	0.83	0.84	0.83	0.51	0.52	0.51	0.71	0.72	0.71
Annotowatch 3	0.44	0.58	0.50	0.39	0.25	0.31	0.37	0.45	0.41	0.83	0.84	0.83	0.51	0.53	0.51	0.70	0.72	0.71
Illinois NER	0.46	0.57	0.51	0.05	0.08	0.07	0.26	0.36	0.30	0.86	0.82	0.84	0.41	0.46	0.43	0.64	0.69	0.66
Stanford NER	0.46	0.60	0.52	0.01	0.01	0.01	0.25	0.31	0.28	0.83	0.80	0.82	0.39	0.43	0.41	0.65	0.66	0.66
Open Calais	0.75	0.52	0.61	0.54	0.20	0.29	0.62	0.19	0.30	0.66	0.73	0.69	0.64	0.41	0.47	0.66	0.60	0.63
Dummy	0.29	0.72	0.41	0.09	0.35	0.15	0.32	0.62	0.42	0.63	0.92	0.75	0.33	0.65	0.43	0.47	0.82	0.60
ANNIE	0.47	0.49	0.48	-	-	-	0.23	0.17	0.19	0.72	0.64	0.68	0.61	0.32	0.34	0.63	0.52	0.57
Illinois Wikifier	0.28	0.44	0.34	0.07	0.13	0.09	0.53	0.41	0.46	0.88	0.55	0.67	0.44	0.38	0.39	0.63	0.49	0.55
Apache OpenNLP	0.36	0.41	0.38	-	-	-	0.14	0.12	0.13	0.78	0.54	0.64	0.57	0.27	0.29	0.62	0.43	0.51
Wikipedia Miner	0.25	0.50	0.33	0.03	0.18	0.05	0.29	0.38	0.33	0.76	0.58	0.66	0.33	0.41	0.34	0.41	0.52	0.46
LingPipe	0.09	0.45	0.15	-	-	-	0.03	0.22	0.05	0.34	0.44	0.38	0.37	0.28	0.14	0.15	0.38	0.21

This work is supported by projects VEGA 2/0185/13, VENIS FP7-284984, CLAN APVV-0809-11 and ITMS: 26240220072