Ensemble Learning for Named Entity Recognition

René Speck and Axel-Cyrille Ngonga Ngomo

AKSW, Department of Computer Science, University of Leipzig, Germany
{speck,ngonga}@informatik.uni-leipzig.de

Abstract. A considerable portion of the information on the Web is still only available in unstructured form. Implementing the vision of the Semantic Web thus requires transforming this unstructured data into structured data. One key step during this process is the recognition of named entities. Previous works suggest that ensemble learning can be used to improve the performance of named entity recognition tools. However, no comparison of the performance of existing supervised machine learning approaches on this task has been presented so far. We address this research gap by presenting a thorough evaluation of named entity recognition based on ensemble learning. To this end, we combine four different state-of-the approaches by using 15 different algorithms for ensemble learning and evaluate their performace on five different datasets. Our results suggest that ensemble learning can reduce the error rate of state-of-the-art named entity recognition systems by 40%, thereby leading to over 95% f-score in our best run.

Keywords: Named Entity Recognition • Ensemble Learning • Semantic Web

1 Introduction

One of the first research papers in the field of named entity recognition (NER) was presented in 1991 [32]. Today, more than two decades later, this research field is still highly relevant for manifold communities including Semantic Web Community, where the need to capture and to translate the content of natural language (NL) with the help of NER tools arises in manifold semantic applications [15, 19, 20, 24, 34]. The NER tools that resulted from more than 2 decades of research now implement a diversity of algorithms that rely on a large number of heterogeneous formalisms. Consequently, these algorithms have diverse strengths and weaknesses.

Currently, several services and frameworks that consume NL to generate semi-structured or even structured data rely on solely one of the formalisms developed for NER or simply merging the results of several tools (e.g., by using simple voting). By doing so, current approaches fail to make use of the diversity of current NER algorithms. On the other hand, it is a well-known fact that algorithms with diverse strengths and weaknesses can be aggregated in various ways to create a system that outperforms the best individual algorithms within the system [44]. This learning paradigm is known as *ensemble learning*. While previous works have already suggested that ensemble learning can be used to improve NER [34], no comparison of the performance of existing supervised machine-learning approaches for ensemble learning on the NER task has been presented so far.

We address this research gap by presenting and evaluating an open-source framework for NER that makes use on ensemble learning. In this evaluation, we use four state-of-the-art NER algorithms, fifteen different machine learning algorithms and five datasets. The statistical significance our results is ensured by using Wilcoxon signed-rank tests.

The goal of our evaluation is to answer the following questions:

1. Does NER based on ensemble learning achieve higher f-scores than the best NER tool within the system?
2. Does NER based on ensemble learning achieve higher f-scores than simple voting based on the results of the NER tools?
3. Which ensemble learning approach achieves the best f-score for the NER task?

The rest of this paper is structured as follows. After reviewing related work in Section 2, we give an overview of our approach in Section 3. Especially, we present the theoretical framework that underlies our approach. Subsequently, in Section 4, we present our evaluation pipeline and its setup. Thereafter, in Section 5, we present the results of a series of experiments in which we compare several machine learning algorithms with state-of-the-art NER tools. We conclude by discussing our results and elaborating on some future work in Section 6. The results of this paper were integrated into the open-source NER framework FOX.[1] Our framework provides a free-to-use RESTful web service for the community. A documentation of the framework as well as a specification of the RESTful web service can be found at FOX's project page.

2 Related Work

NER tools and frameworks implement a broad spectrum of approaches, which can be subdivided into three main categories: dictionary-based, rule-based and machine-learning approaches [31]. The first systems for NER implemented dictionary-based approaches, which relied on a list of named entities (NEs) and tried to identify these in text [2,43]. Following work then showed that these approaches did not perform well for NER tasks such as recognizing proper names [39]. Thus, rule-based approaches were introduced. These approaches rely on hand-crafted rules [8,42] to recognize NEs. Most rule-based approaches combine dictionary and rule-based algorithms to extend the list of known entities. Nowadays, hand-crafted rules for recognizing NEs are usually implemented when no training examples are available for the domain or language to process [32]. When training examples are available, the methods of choice are borrowed from supervised machine learning. Approaches such as Hidden Markov Models [46], Maximum Entropy Models [10] and Conditional Random Fields [14] have been applied to the NER task. Due to scarcity of large training corpora as necessitated by supervised machine learning approaches, the semi-supervised [31, 35] and unsupervised machine learning paradigms [13, 33] have also been used for extracting NER from text. In [44], a system was presented that combines with stacking and voting classifiers which were

[1] Project page:`http://fox.aksw.org`. Source code, evaluation data and evaluation results:`http://github.com/AKSW/FOX`.

trained with several languages, for language-independent NER. [31] gives an exhaustive overview of approaches for the NER task.

Over the last years, several benchmarks for NER have been proposed. For example, [9] presents a benchmark for NER and entity linking approaches. Especially, the authors define the named entity annotation task. Other benchmark datasets include the manually annotated datasets presented in [38]. Here, the authors present annotated datasets extracted from RSS feeds as well as datasets retrieved from news platforms. Other authors designed datasets to evaluate their own systems. For example, the `Web` dataset (which we use in our evaluation) is a particularly noisy dataset designed to evaluate the system presented in [37]. The dataset `Reuters`, which we also use, consists annotated documents chosen out of the Reuters-215788 corpus and was used in [4].

3 Overview

3.1 Named Entity Recognition

NER encompasses two main tasks: (1) The identification of names[2] such as "`Germany`", "`University of Leipzig`" and "`G. W. Leibniz`" in a given unstructured text and (2) the classification of these names into predefined entity types[3], such as `Location`, `Organization` and `Person`. In general the NER task can be viewed as the sequential prediction problem of estimating the probabilities $P(y_i|x_{i-k}...x_{i+l}, y_{i-m}...y_{i-1})$, where $\mathbf{x} = (x_1, .., x_n)$ is an input sequence (i.e., the preprocessed input text) and $\mathbf{y} = (y_1, ..., y_n)$ the output sequence (i.e., the entity types) [37].

3.2 Ensemble Learning

The goal of an ensemble learning algorithm $\mathcal{S}$ is to generate a classifier $\mathcal{F}$ with a high predictive performance by combining the predictions of a set of m basic classifiers $\mathcal{C}_1, \ldots, \mathcal{C}_m$ [12]. One central observation in this respect, is that combining $\mathcal{C}_1, \ldots, \mathcal{C}_m$ can only lead to a high predictive performance when these classifiers are *accurate* and *diverse* [45]. Several approaches have been developed to allow an efficient combination of basic classifiers. The simplest strategy is voting, where each input token is classified as belonging to the class that was predicted by the largest number of basic classifiers [12]. Voting can be extended to weighted voting, where each of the basic classifiers is assigned a weight and $\mathcal{S}$ returns the class with the highest total prediction weight. More elaborate methods try to ensure the diversity of the classifiers. Approaches that aim to achieve this goal include drawing random samples (with replacement) from the training data (e.g., bagging, [5]) or generating sequences of classifiers of high diversity that are trained to recognized each other's mistakes (e.g., boosting, [40]). The results of all classifiers are finally combined via weighted voting.

Here, we consider ensemble learning for NER. Thanks to the long research tradition on the NER topic, the *diversity* and *accuracy* of the tools is already available and can be regarded as given. However, classical ensemble learning approaches present the

[2] Also referred as instances.

[3] Also referred as classes.

disadvantage of relying on some form of weighted vote on the output of the classifiers. Thus, if all classifiers $\mathcal{C}_i$ return wrong results, classical ensemble learning approaches are bound to make the same mistake [12]. In addition, voting does not take the different levels of accuracy of classifiers for different entity types into consideration. Rather, it assigns a global weight to each classifier that describes its overall accuracy. Based on these observations, we decided to apply ensemble learning for NER based at entity-type level. The main advantage of this ensemble-learning setting is that we can now assign different weights to each tool-type pair.

Formally, we model the ensemble learning task at hand as follows: Let the matrix $M^{mt \times n}$ (Equation 1) illustrate the input data for $\mathcal{S}$, where $\mathcal{P}^m_{n,t}$ are predictions of the m-th NER tool that the n-th token is of the t-th type.

$$\begin{pmatrix} \mathcal{P}^1_{1,1} & \cdots & \mathcal{P}^1_{1,t} & \mathcal{P}^2_{1,1} & \cdots & \mathcal{P}^2_{1,t} & \cdots & \mathcal{P}^m_{1,1} & \cdots & \mathcal{P}^m_{1,t} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \mathcal{P}^1_{n,1} & \cdots & \mathcal{P}^1_{n,t} & \mathcal{P}^2_{n,1} & \cdots & \mathcal{P}^2_{n,t} & \cdots & \mathcal{P}^m_{n,1} & \cdots & \mathcal{P}^m_{n,t} \end{pmatrix} \tag{1}$$

The goal of ensemble learning for NER is to detect a classifier that leads to a correct classification of each of the n tokens into one of the types t.

4 Evaluation

We performed a thorough evaluation of ensemble learning approaches by using five different datasets and running a 10-fold cross-validation for 15 algorithms. In this section, we present the pipeline and the setup for our evaluation as well as our results.

4.1 Pipeline

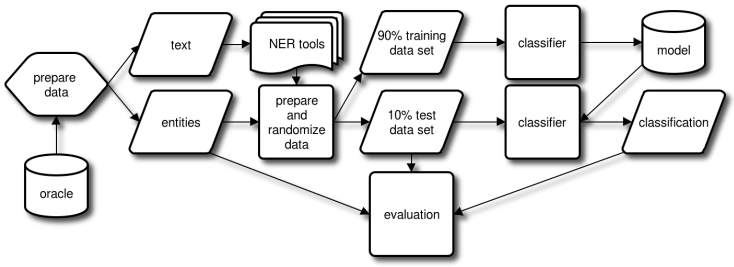


Fig. 1: Workflow chart of the evaluation pipeline.

Figure 1 shows the workflow chart of our evaluation pipeline. In the first step of our evaluation pipeline, we preprocessed our reference dataset to extract the input text for the NER tools as well as the correct NEs, which we used to create training and testing data. In the second step, we made use of all NER tools with this input text to

calculate the predictions of all entity types for each token in this input. At this point, we represented the output of the tools as matrix (see Equation 1). Thereafter, the matrix was randomly split into 10 disjoint sets as preparation for a 10-fold cross-validation. We trained the different classifiers at hand (i.e., $\mathcal{S}$) with the training dataset (i.e., with 9 of 10 sets) and tested the trained classifier with the testing dataset (i.e., with the leftover set). To use each of the 10 sets as testing set once, we repeated training and testing of the classifiers 10 times and used the disjoint sets accordingly. Furthermore, the pipeline was repeated 10 times to deal with non-deterministic classifiers. In the last step, we compared the classification of the 10 testing datasets with the oracle dataset to calculate measures for the evaluation.

We ran our pipeline on 15 ensemble learning algorithms. We carried out both a token-based evaluation and an entity-based evaluation. In the *token-based evaluation*, we regarded partial matches of multi-word units as being partially correct. For example, our gold standard considered "`Federal Republic of Germany`" as being an instance of `Location`. If a tool generated "`Germany`" as being a location and omitted "`Federal Republic of`", it was assigned 1 true positive and 3 false negatives. The *entity-based evaluation* only regarded exact matches as correct. In the example above, the entity was simply considered to be incorrect. To provide transparent results, we only used open-source libraries in our evaluation. Given that some of these tools at hand do not allow accessing their confidence score without any major alteration of their code, we considered the output of the tools to be binary (i.e., either 1 or 0).

We integrated four NER tools so far: the Stanford Named Entity Recognizer[4] (Stanford) [14], the Illinois Named Entity Tagger[5] (Illinois) [37], the Ottawa Baseline Information Extraction[6] (Balie) [30] and the Apache OpenNLP Name Finder[7] (OpenNLP) [3]. We only considered the performance of these tools on the classes `Location`, `Organization` and `Person`. To this end, we mapped the entity types of each of the NER tools to these three classes. We utilized the Waikato Environment for Knowledge Analysis (Weka) [21] and the implemented classifiers with default parameters: AdaBoostM1 (ABM1) [16] and Bagging (BG) [5] with J48 [36] as base classifier, Decision Table (DT) [26], Functional Trees (FT) [18, 27], J48 [36], Logistic Model Trees (LMT) [27, 41], Logistic Regression (Log) [28], Additive Logistic Regression (LogB) [17], Multilayer Perceptron (MLP), Naïve Bayes (NB) [23], Random Forest (RF) [6], Support Vector Machine (SVM) [7] and Sequential Minimal Optimization (SMO) [22]. In addition, we used voting at class level (CVote) and a simple voting (Vote) approach [44] with equal weights for all NER tools. CVote selects the NER tool with the highest prediction performance for each type according to the evaluation and uses that particular tool for the given class. Vote as naive approach combines the results of the NER tools with the Majority Vote Rule [25] and was the baseline ensemble learning technique in our evaluation.

[4] `http://nlp.stanford.edu/software/CRF-NER.shtml` (version 3.2.0)

[5] `http://cogcomp.cs.illinois.edu/page/software_view/NETagger` (version 2.4.0)

[6] `http://balie.sourceforge.net` (version 1.8.1)

[7] `http://opennlp.apache.org/index.html` (version 1.5.3)

4.2 Experimental Setup

We used five datasets and five measures for our evaluation. We used the recommended Wilcoxon signed-rank test to measure the statistical significance of our results [11]. For this purpose, we applied each measurement of the ten 10-fold cross-validation runs for the underlying distribution and we set up a 95% confidence interval.

Datasets An overview of the datasets is shown in Table 1. The `Web` dataset consists of 20 annotated Web sites as described in [37] and contains the most noise compared to the other datasets. The dataset `Reuters` consists of 50 documents randomly chosen out of the Reuters-215788 corpus[8] [4]. `News`* is a small subset of the dataset `News` that consists of text from newspaper articles and was re-annotated manually by the authors to ensure high data quality. Likewise, `Reuters` was extracted and annotated manually by the authors. The last dataset, `All`, consists of the datasets mentioned before merged into one and allows for measuring how well the ensemble learning approaches perform when presented with data from heterogenous sources.

Table 1: Number of entities separated according entity types and in total.

Class	News	News*	Web	Reuters	All
`Location`	5117	341	114	146	5472
`Organization`	6899	434	257	208	7467
`Person`	3899	254	396	91	4549
Total	15915	1029	767	445	17488

Measures To assess the performance of the different algorithms, we computed the following values on the test datasets: The number of true positives TP_t, the number of true negatives TN_t, the number of false positives FP_t and the number of false negatives FN_t. These numbers were collected for each entity type t and averaged over the ten runs of the 10-fold cross-validations. Then, we applied the one-against-all approach [1] to convert the multi-class confusion matrix of each dataset into a binary confusion matrix.

Subsequently, we determined with macro-averaging the classical measures recall (rec), precision (pre) and f-score (F_1) as follows:

$$rec = \frac{\sum\limits_{t \in T} \frac{TP_t}{(TP_t + FN_t)}}{|T|}, pre = \frac{\sum\limits_{t \in T} \frac{TP_t}{TP_t + FP_t}}{|T|}, F_1 = \frac{\sum\limits_{t \in T} \frac{2 pre_t rec_t}{pre_t + rec_t}}{|T|}. \quad (2)$$

[8] The Reuters-215788 corpus is available at:
`http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`.

For the sake of completeness, we averaged the error rate ($error$) (Equation 3) and the Matthews correlation coefficient (MCC) [29] (Equation 4) similarly.

$$error = \frac{\sum\limits_{t \in T} \frac{FP_t + FN_t}{TP_t + TN_t + FP_t + FN_t}}{|T|} \tag{3}$$

$$MCC = \frac{\sum\limits_{t \in T} \frac{TP_t TN_t - FP_t FN_t}{\sqrt{(TP_t + FP_t)(TP_t + FN_t)(TN_t + FP_t)(TN_t + FN_t)}}}{|T|} \tag{4}$$

The error rate monitors the fraction of positive and negative classifications for that the classifier failed. The Matthews correlation coefficient considers both the true positives and the true negatives as successful classification and is rather unaffected by sampling biases. Higher values indicating better classifications.

5 Results

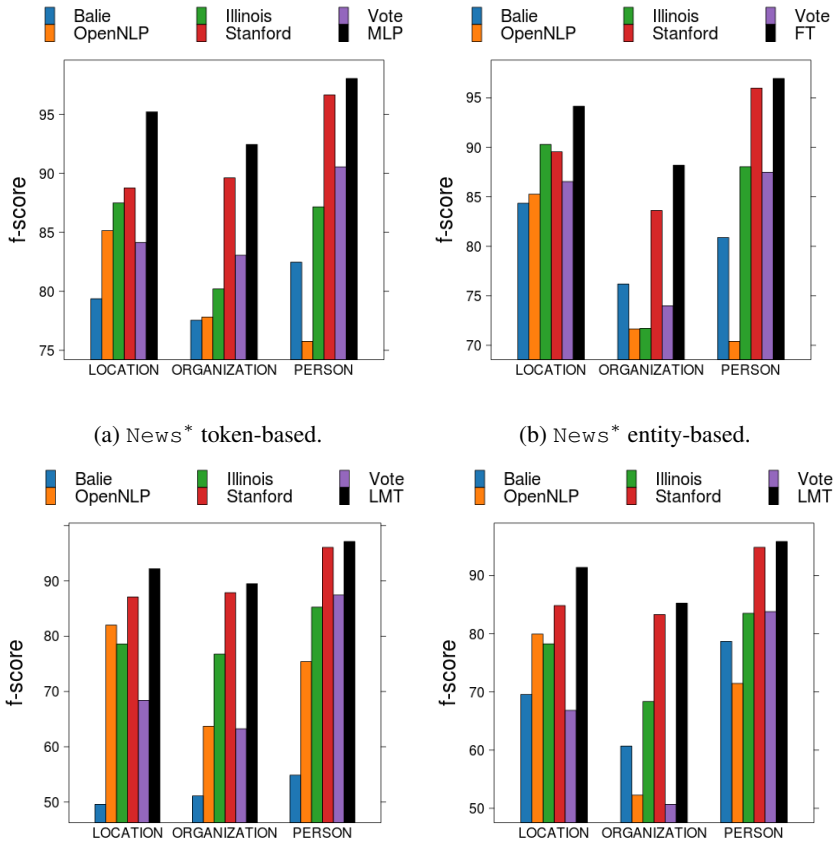
Table 2–Table 11 show the results of our evaluation for the 15 classifiers we used within our pipeline and the four NER tools we integrated so far. The best results are marked bold and the NER tools are underlined. Figure 2–Figure 4 depict the f-scores separated according classes of the four NER tools, the simple voting approach Vote and the best classifier for the depicted dataset.

Table 2: `News*` token-based.

$\mathcal{S}$	rec	pre	F_1	$error$	MCC
MLP	**95.19**	**95.28**	**95.23**	**0.32**	**0.951**
RF	95.15	**95.28**	95.21	**0.32**	**0.951**
ABM1	94.82	95.18	95.00	0.33	0.948
SVM	94.86	95.09	94.97	0.33	0.948
J48	94.78	94.98	94.88	0.34	0.947
BG	94.76	94.93	94.84	0.34	0.947
LMT	94.68	94.95	94.82	0.34	0.946
DT	94.63	94.95	94.79	0.34	0.946
FT	94.30	95.15	94.72	0.35	0.945
LogB	93.54	95.37	94.44	0.37	0.943
Log	94.05	94.75	94.40	0.37	0.942
SMO	94.01	94.37	94.19	0.39	0.940
NB	94.61	92.64	93.60	0.42	0.934
Stanford	92.36	91.01	91.68	0.53	0.914
CVote	92.02	90.84	91.42	0.54	0.911
Vote	89.98	82.97	85.92	0.94	0.857
Illinois	82.79	87.35	84.95	0.92	0.845
Balie	77.68	82.05	79.80	1.21	0.792
OpenNLP	71.42	90.47	79.57	1.13	0.797

Table 3: `News*` entity-based.

$\mathcal{S}$	rec	pre	F_1	$error$	MCC
FT	93.95	**92.27**	**93.10**	**0.30**	**0.930**
MLP	**94.10**	92.13	93.09	**0.30**	0.929
LMT	94.08	91.91	92.97	0.31	0.928
RF	93.76	92.07	92.90	0.31	0.928
BG	93.51	92.18	92.83	0.31	0.927
SVM	93.85	91.46	92.62	0.32	0.925
ABM1	93.30	91.65	92.47	0.33	0.923
J48	93.30	91.65	92.47	0.33	0.923
Log	93.42	91.39	92.37	0.33	0.922
LogB	92.89	91.68	92.27	0.33	0.921
SMO	92.55	91.26	91.90	0.36	0.917
DT	92.44	91.29	91.86	0.34	0.917
NB	94.08	88.26	91.01	0.40	0.909
Stanford	92.00	87.58	89.72	0.45	0.895
CVote	91.43	86.94	89.10	0.47	0.889
Illinois	82.07	84.84	83.34	0.67	0.831
Vote	91.42	76.52	82.67	0.83	0.829
Balie	81.54	79.66	80.48	0.79	0.801
OpenNLP	69.36	85.02	75.78	0.88	0.760



(a) `News`* token-based.

(b) `News`* entity-based.

(c) `News` token-based.

(d) `News` entity-based.

Fig. 2: `News` and `News`* dataset.

We reached the highest f-scores on the `News`* dataset (Table 2 and Table 3) for both the token-based and the entity-based evaluation. In the token-based evaluation, the MLP and RF classifiers perform best for precision (95.28%), error rate (0.32%) and Matthews correlation coefficient (0.951). MLP performs best for f-score (95.23%) with 0.04% more recall than RF. The baseline classifier (i.e., simple voting) is clearly outperformed by MLP by up to +5.21% recall, +12.31% precision, +9.31% f-score, -0.62% error rate and +0.094 MCC. Furthermore, the best single approach is Stanford and outperformed by up to +2.83% recall, +4.27% precision, +3.55% f-score, -0.21% error rate (that is a reduction by 40%) and +0.037 MCC. Slightly poorer results are achieved in the entity-based evaluation, where MLP is second to FT with 0.01% less f-score.

On the `News` dataset (Table 4-Table 5), which was the largest homogenous dataset in our evaluation, we repeatedly achieved high f-scores. The best approach w.r.t. the

Table 4: `News` token-based.

$\mathcal{S}$	rec	pre	F_1	$error$	MCC
LMT	**93.73**	92.16	**92.94**	**0.51**	**0.927**
RF	93.56	92.19	92.87	**0.51**	0.926
DT	93.64	92.10	92.86	**0.51**	0.926
J48	93.50	92.20	92.84	0.52	0.926
ABM1	93.49	92.17	92.83	0.52	0.926
BG	93.11	**92.49**	92.79	0.52	0.925
FT	93.44	92.15	92.79	0.52	0.925
MLP	93.22	92.26	92.73	0.52	0.925
SVM	92.19	**92.49**	92.31	0.54	0.920
SMO	92.15	91.90	92.01	0.57	0.917
Log	91.38	91.36	91.35	0.63	0.910
LogB	91.42	91.32	91.34	0.62	0.910
Stanford	92.70	88.09	90.34	0.68	0.900
CVote	92.70	88.09	90.34	0.68	0.900
NB	93.36	86.17	89.58	0.77	0.893
Illinois	82.43	78.11	80.20	1.37	0.795
OpenNLP	75.21	74.41	73.71	2.06	0.732
Vote	83.13	69.14	73.03	2.36	0.735
Balie	70.81	72.86	71.54	1.90	0.707

Table 5: `News` entity-based.

$\mathcal{S}$	rec	pre	F_1	$error$	MCC
LMT	**92.95**	88.84	**90.84**	**0.44**	**0.906**
BG	92.82	88.95	90.83	**0.44**	**0.906**
DT	92.89	88.88	90.83	**0.44**	**0.906**
ABM1	92.87	88.82	90.79	**0.44**	**0.906**
J48	92.87	88.82	90.79	**0.44**	**0.906**
FT	92.90	88.78	90.78	**0.44**	**0.906**
RF	92.84	88.77	90.74	**0.44**	**0.906**
MLP	92.83	88.69	90.70	**0.44**	0.905
SVM	91.56	**89.22**	90.33	0.45	0.901
SMO	91.13	88.36	89.69	0.49	0.895
Log	90.62	88.09	89.29	0.51	0.891
LogB	90.76	87.83	89.22	0.51	0.890
Stanford	91.78	83.92	87.66	0.58	0.875
CVote	91.78	83.92	87.66	0.58	0.875
NB	92.54	81.16	86.34	0.69	0.863
Illinois	81.66	72.50	76.71	1.11	0.763
Balie	71.58	68.67	69.66	1.42	0.692
OpenNLP	72.71	67.29	67.89	1.80	0.681
Vote	82.71	61.30	67.10	2.19	0.686

token-based evaluation is LMT with an f-score of 92.94%. Random Forest follows the best approach with respect to f-score again. Moreover, the best single tool Stanford and the baseline classifier Vote are repeatedly outperformed by up to +2.6% resp. +19.91% f-score. Once again, the entity-based results are approximately 2% poorer, with LMT leading the table like in the token-based evaluation.

On the `Web` dataset (Table 6-Table 7), which is the worst-case dataset for NER tools as it contains several incomplete sentences, the different classifiers reached their lowest values. For the token-based evaluation, AdaBoostM1 with J48 achieves the best f-score (69.04%) and Matthews correlation coefficient (0.675) and is followed by Random Forest again with respect to f-score. Naïve Bayes performs best for recall (96.64%), Logistic Regression for precision (77.89%) and MLP and RF for the error rate (3.33%). Simple voting is outperformed by ABM1 by up to +3.5% recall, +20.08% precision, +10.45% f-score, -2.64% error rate and +0.108 MCC, while Stanford (the best tool for this dataset) is outperformed by up to +3.83% recall, +2.64% precision, +3.21% f-score, -0.13% error rate and +0.032 MCC. Similar insights can be won from the entity-based evaluation, with some classifiers like RF being approximately 10% poorer that at token level.

On the `Reuters` dataset (Table 8-Table 9), which was the smallest dataset in our evaluation, Support Vector Machine performs best. In the token-based evaluation, SVM achieves an f-score of 87.78%, an error rate of 0.89% and a Matthews correlation coefficient of 0.875%. They are followed by Random Forest with respect to f-score once again. Naïve Bayes performs best for recall (86.54%). In comparison, ensemble learning outperforms Vote with SVM by up to +4.46% recall, +3.48% precision, +2.43% f-

Table 6: `Web` token-based.

$\mathcal{S}$	rec	pre	F_1	$error$	MCC
ABM1	64.40	74.83	**69.04**	3.38	**0.675**
RF	64.36	74.57	68.93	3.38	0.674
MLP	63.86	75.11	68.81	**3.33**	0.674
FT	62.98	75.47	68.25	**3.33**	0.670
LMT	63.39	74.24	68.04	3.43	0.666
DT	62.80	74.18	67.85	3.43	0.664
CVote	63.16	73.54	67.66	3.49	0.662
SVM	62.94	73.45	67.60	3.49	0.661
LogB	60.47	77.48	67.57	3.40	0.665
Log	60.31	**77.89**	67.50	3.39	0.666
SMO	63.47	72.45	67.49	3.57	0.659
BG	61.06	76.19	67.46	3.34	0.663
J48	62.21	73.78	67.21	3.49	0.658
NB	**71.19**	63.42	66.88	4.42	0.647
Stanford	60.57	72.19	65.81	3.51	0.643
Illinois	69.64	60.56	64.44	5.09	0.621
Vote	66.90	54.75	58.59	6.02	0.567
OpenNLP	45.71	58.81	49.18	5.93	0.477
Balie	38.63	43.83	40.15	7.02	0.371

Table 7: `Web` entity-based.

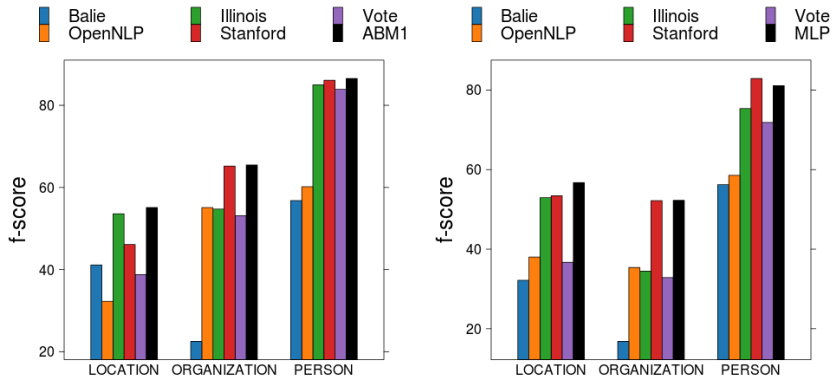
$\mathcal{S}$	rec	pre	F_1	$error$	MCC
MLP	64.95	61.86	**63.36**	1.99	**0.624**
Stanford	64.80	61.31	62.83	1.95	0.619
LogB	61.25	**64.10**	62.60	**1.94**	0.616
FT	63.67	61.10	62.21	2.09	0.612
ABM1	63.49	61.01	62.17	2.08	0.611
Log	60.43	63.62	61.95	1.99	0.610
CVote	65.69	59.54	61.82	2.05	0.612
J48	63.21	59.72	61.39	2.12	0.603
BG	64.04	59.10	61.30	2.13	0.603
RF	64.15	55.88	59.69	2.27	0.587
SVM	62.36	57.26	59.57	2.15	0.586
DT	61.92	57.05	59.34	2.17	0.583
LMT	61.25	56.89	58.96	2.19	0.579
SMO	62.44	56.01	58.83	2.21	0.579
NB	**74.18**	49.20	58.55	3.17	0.586
Illinois	69.31	45.85	54.25	3.82	0.541
Vote	67.42	37.77	47.12	4.84	0.477
OpenNLP	46.94	46.78	43.99	3.71	0.437
Balie	38.07	32.92	35.07	3.63	0.334

Table 8: `Reuters` token-based.

$\mathcal{S}$	rec	pre	F_1	$error$	MCC
SVM	84.57	**91.75**	**87.78**	**0.89**	**0.875**
RF	86.11	89.24	87.58	0.90	0.872
MLP	85.89	89.46	87.55	0.90	0.871
LMT	84.41	91.08	87.43	**0.89**	0.871
J48	84.64	90.70	87.33	0.93	0.870
Log	84.33	90.85	87.27	**0.89**	0.870
LogB	84.22	91.01	87.22	0.90	0.870
ABM1	84.51	90.47	87.15	0.93	0.868
BG	84.70	90.16	87.14	0.94	0.868
FT	85.25	88.75	86.87	0.95	0.864
DT	84.41	89.00	86.43	0.99	0.861
SMO	84.45	88.49	86.28	0.98	0.859
Illinois	83.74	88.27	85.35	1.09	0.851
NB	**86.54**	83.18	84.77	1.10	0.842
CVote	81.96	88.66	84.64	1.14	0.844
Stanford	81.57	84.85	82.85	1.20	0.824
Vote	80.11	81.15	79.41	1.43	0.793
OpenNLP	67.94	82.08	73.96	1.76	0.736
Balie	64.92	68.61	64.78	2.62	0.645

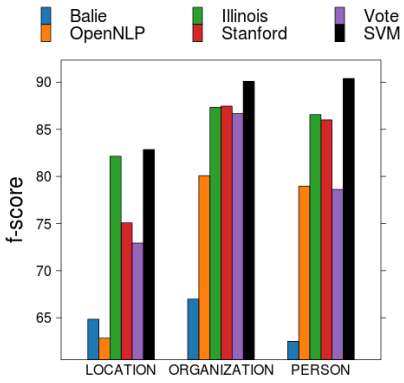
Table 9: `Reuters` entity-based.

$\mathcal{S}$	rec	pre	F_1	$error$	MCC
SVM	81.37	**88.85**	**84.71**	**0.69**	**0.846**
ABM1	80.60	88.72	84.15	0.73	0.840
LMT	80.80	87.92	83.96	0.73	0.838
J48	80.41	88.50	83.95	0.73	0.838
BG	80.55	87.70	83.75	0.75	0.836
Illinois	82.77	85.73	83.74	0.72	0.836
LogB	80.70	86.23	83.32	0.75	0.830
DT	81.11	85.20	82.95	0.79	0.827
RF	80.08	86.11	82.86	0.78	0.826
Log	80.01	85.51	82.62	0.78	0.823
MLP	80.27	84.09	81.98	0.83	0.817
SMO	79.62	83.21	81.36	0.88	0.809
FT	80.00	82.71	81.32	0.85	0.809
CVote	77.86	85.42	81.00	0.85	0.809
NB	**83.80**	77.68	80.61	0.92	0.802
Stanford	77.56	82.38	79.68	0.90	0.794
Vote	80.35	76.25	77.37	1.03	0.773
OpenNLP	66.85	80.33	72.89	1.18	0.726
Balie	68.90	70.14	68.71	1.39	0.684

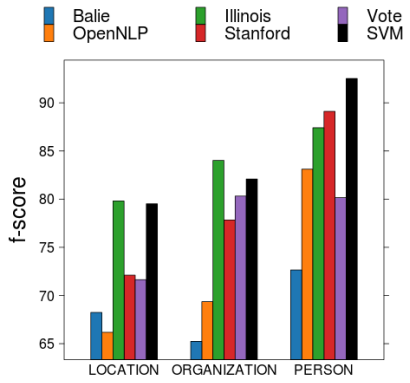


(a) `Web` token-based.

(b) `Web` entity-based.



(c) `Reuters` token-based.



(d) `Reuters` entity-based.

Fig. 3: `Web` and `Reuters` dataset.

score, -0.54% error rate and +0.082 MCC. Moreover, the best NER tool for this dataset, Illinois, is outperformed by up to +0.83% recall, +3.48% precision, +2.43% f-score, -0.20% error rate and +0.024 MCC. In Figure 3a, we barely see a learning effect as ABM1 is almost equal to one of the integrated NER tools assessed at class level especially for the class `Organization` on the `Web` dataset but in Figure 3c on the `Reuters` dataset we clearly see a learning effect for the class `Organization` and `Person` with the SVM approach.

On the `All` dataset for token-based evaluation (Table 10), the Random Forest approach performs best for f-score (91.27%), error rate (0.64%) and Matthews correlation coefficient (0.909). Support Vector Machine achieves the best precision (91.24%) and Naïve Bayes the best recall (91.00%) again. In comparison, ensemble learning outperformed Vote with RF by up to +9.71% recall, +21.01% precision, +18.37% f-score, -1.8% error rate and +0.176% MCC and Stanford, the best tool for this dataset, by up

Table 10: `All` token-based.

$\mathcal{S}$	rec	pre	F_1	$error$	MCC
RF	91.58	90.97	**91.27**	**0.64**	**0.909**
LMT	91.67	90.86	91.26	**0.64**	**0.909**
ABM1	91.49	90.99	91.24	**0.64**	**0.909**
J48	91.46	90.98	91.22	**0.64**	**0.909**
DT	91.59	90.84	91.21	**0.64**	**0.909**
FT	91.49	90.82	91.16	0.65	0.908
BG	91.25	91.00	91.12	0.65	0.908
MLP	90.94	91.05	90.99	0.66	0.907
SVM	90.15	**91.24**	90.67	0.67	0.903
SMO	90.13	90.48	90.27	0.71	0.899
Log	88.69	90.57	89.59	0.76	0.892
LogB	88.92	90.21	89.53	0.76	0.892
Stanford	90.75	87.73	89.21	0.78	0.888
CVote	90.75	87.73	89.21	0.78	0.888
NB	**92.00**	85.27	88.46	0.89	0.881
Illinois	81.66	77.61	79.54	1.48	0.788
Vote	81.85	69.96	72.90	2.44	0.733
OpenNLP	72.63	75.60	72.65	2.19	0.723
Balie	67.75	71.65	69.40	2.09	0.685

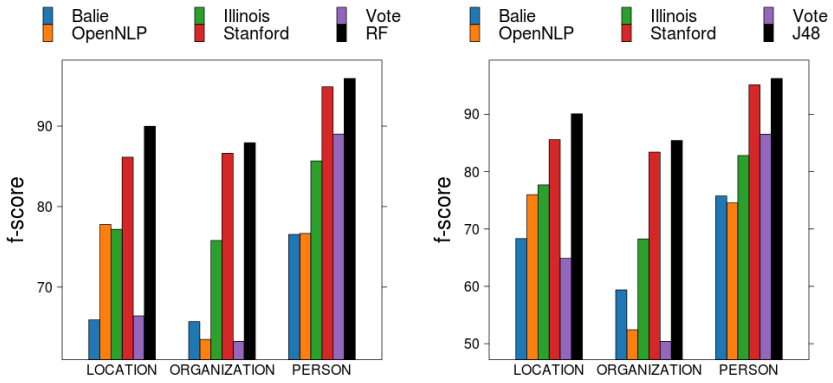
Table 11: `All` entity-based.

$\mathcal{S}$	rec	pre	F_1	$error$	MCC
J48	92.68	88.62	**90.59**	**0.44**	**0.904**
ABM1	92.66	88.59	90.56	**0.44**	**0.904**
LMT	92.59	88.50	90.48	0.45	0.903
DT	92.56	88.44	90.44	0.45	0.902
RF	92.51	88.33	90.35	0.45	0.902
FT	92.47	88.37	90.35	0.45	0.902
BG	92.17	88.55	90.31	0.45	0.901
MLP	92.07	88.60	90.28	0.45	0.901
SVM	90.91	**88.97**	89.88	0.46	0.897
SMO	90.94	87.31	89.00	0.52	0.888
Log	89.49	88.10	88.70	0.53	0.885
LogB	89.21	87.68	88.36	0.54	0.881
Stanford	92.00	84.48	88.05	0.56	0.879
CVote	92.00	84.48	88.05	0.56	0.879
NB	**92.69**	80.59	86.04	0.71	0.860
Illinois	81.43	71.82	76.25	1.12	0.759
Balie	69.27	67.47	67.82	1.48	0.674
OpenNLP	71.29	69.44	67.66	1.80	0.682
Vote	81.97	62.17	67.27	2.17	0.687

to +0.83% recall, +3.24% precision, +2.06% f-score, -0.14% error rate and +0.021% MCC. Again, entity-based evaluation (Table 11) compared to token-based evaluation, the f-score of J48, the best ensemble learning approach here, is approximately 1% poorer with higher recall but lower precision. In Figure 4, we clearly see a learning effect for RF and J48 at class level.

Overall, ensemble learning outperform all included NER tools and the simple voting approach for all datasets with respect to f-score, which answers our first and second question. Here, it is worth mentioning that Stanford and Illinois are the best tools in our framework. The three best classifiers with respect to the averaged f-scores over our datasets for token-based evaluation are the Random Forest classifier with the highest value, closely followed by Multilayer Perceptron and AdaBoostM1 with J48 and for entity-based evaluation AdaBoostM1 with J48 with the highest value, closely followed by MLP and J48. We cannot observe a significant difference between these.

In Table 12 and Table 13, we depict the f-scores of these three classifiers at class level for our datasets. The statistically significant differences are marked in bold. Note that two out of three scores being marked bold for the same setting in a column means that the corresponding approaches are significantly better than the third one yet not significantly better than each other. In the token-based evaluation, the Multilayer Perceptron and Random Forest classifier surpass the AdaBoostM1 with J48 on the `News`* and `Web` datasets. On the `News`* dataset, MLP surpasses RF for `Location` but RF surpasses MLP for `Person`. On the `Web` dataset, RF is better than MLP for `Location` but not significantly different from one another for `Person`. Also, for the `Organization` class, no significant difference could be determined on both datasets. On the `Reuters`



(a) `All` token-based.

(b) `All` entity-based.

Fig. 4: `All` dataset.

dataset, MLP and RF are better than ABM1 for `Location` and `Organization`, but do not differ one another. For the class `Person`, no significant difference could be determined for all three classifiers. On the `News` and `All` dataset, Random Forest is significantly best for `Location`. Random Forest and AdaBoostM1 with J48 surpass the Multilayer Perceptron for `Organization` but are not significantly different. For the class `Person`, ABM1 is significantly best on the `News` dataset and RF is best on the `All` dataset. The entity-level results also suggest shifts amongst the best systems depending on the datasets. Interestingly, MLP and ABM1 are the only two classes of algorithm that appear as top algorithms in both evaluation schemes.

Consequently, our results suggest that while the four approaches RF, MLP, ABM1 and J48 perform best over the datasets at hand, MLP and ABM1 are to be favored. Note that significant differences can be observed across the different datasets and that all four paradigms RF, MLP, ABM1 and J48 should be considered when applying ensemble learning to NER. This answers the last and most important question of this evaluation.

Table 12: F-score of the best 3 classifiers on class level token-based.

$\mathcal{S}$	Class	News	News*	Web	Reuters	All
RF	`Location`	**92.12**	94.96	**54.58**	**82.25**	**89.98**
RF	`Organization`	**89.45**	**92.44**	**65.60**	**90.53**	**87.93**
RF	`Person`	97.02	**98.25**	**86.61**	89.95	**95.91**
MLP	`Location`	91.79	**95.22**	53.78	**82.13**	89.62
MLP	`Organization`	89.34	**92.45**	**65.72**	**90.38**	87.63
MLP	`Person`	**97.07**	98.04	**86.94**	90.14	95.73
ABM1	`Location`	91.75	95.10	55.11	81.19	89.90
ABM1	`Organization`	**89.49**	92.00	65.47	89.91	**87.96**
ABM1	`Person`	**97.12**	97.89	86.53	90.37	95.87

Table 13: F-score of the best 3 classifiers on class level entity-based.

$\mathcal{S}$	Class	News	News*	Web	Reuters	All
ABM1	`Location`	91.26	95.71	**58.21**	**78.99**	90.05
ABM1	`Organization`	**85.19**	85.87	50.66	**80.45**	85.43
ABM1	`Person`	**95.91**	95.81	77.63	**93.02**	96.21
MLP	`Location`	91.14	95.35	56.72	76.32	89.63
MLP	`Organization`	85.17	**87.30**	**52.29**	78.74	85.38
MLP	`Person`	95.79	96.61	**81.09**	90.88	95.83
J48	`Location`	**91.27**	95.71	56.53	**78.99**	**90.08**
J48	`Organization`	85.18	85.87	50.56	**80.49**	85.44
J48	`Person`	**95.91**	95.81	77.10	92.36	**96.23**

6 Conclusion and Future Work

In this paper, we evaluated named entity recognition based on ensemble learning, an approach to increase the performance of state-of-the-art named entity recognition tools. On all datasets, we showed that ensemble learning achieves higher f-scores than the best named entity recognition tool integrated in our system and higher f-scores compared with a simple voting on the outcome of the integrated tools. Our results suggest that Multilayer Perceptron and AdaBoostM1 with J48 as base classifier work best for the task at hand. We have now integrated the results of this evaluation into the FOX framework, which can be found at `http://fox.aksw.org`. The main advantages of our framework are that it is not limited to the integration of named entity recognition tools or ensemble learning algorithms and can be easily extended. Moreover, it provides additional features like linked data and a RESTful web service to use by the community.

References

1. Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *J. Mach. Learn. Res.*, 1:113–141, September 2001.
2. R. Amsler. Research towards the development of a lexical knowledge base for natural language processing. *SIGIR Forum*, 23:1–2, 1989.
3. J Baldridge. The opennlp project, 2005.
4. S. D. Bay and S. Hettich. The UCI KDD Archive [http://kdd.ics.uci.edu], 1999.
5. Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
6. Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
7. Chih-Chung Chang and Chih-Jen Lin. Libsvm - a library for support vector machines, 2001. The Weka classifier works with version 2.82 of LIBSVM.
8. Sam Coates-Stephens. The analysis and acquisition of proper names for the understanding of free text. *Computers and the Humanities*, 26:441–456, 1992. 10.1007/BF00136985.
9. Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 249–260. International World Wide Web Conferences Steering Committee, 2013.

10. James R. Curran and Stephen Clark. Language independent ner using a maximum entropy tagger. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, pages 164–167, 2003.
11. Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, December 2006.
12. Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 1–15, London, UK, 2000. Springer-Verlag.
13. Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165:91–134, June 2005.
14. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, pages 363–370, 2005.
15. Nuno Freire, José Borbinha, and Pável Calado. An approach for named entity recognition in poorly structured data. In Elena Simperl, Philipp Cimiano, Axel Polleres, Oscar Corcho, and Valentina Presutti, editors, *The Semantic Web: Research and Applications*, volume 7295 of *Lecture Notes in Computer Science*, pages 718–732. Springer Berlin Heidelberg, 2012.
16. Yoav Freund and Robert E. Schapire. Experiments with a New Boosting Algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.
17. J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Stanford University, 1998.
18. Joao Gama. Functional trees. 55(3):219–250, 2004.
19. Aldo Gangemi. A comparison of knowledge extraction tools for the semantic web. In Philipp Cimiano, Óscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *ESWC*, volume 7882 of *Lecture Notes in Computer Science*, pages 351–366. Springer, 2013.
20. Sherzod Hakimov, Salih Atilay Oto, and Erdogan Dogdu. Named entity recognition and disambiguation using linked data and graph-based centrality scoring. In *Proceedings of the 4th International Workshop on Semantic Web Information Management*, SWIM '12, pages 4:1–4:7, New York, NY, USA, 2012. ACM.
21. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.
22. Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1998.
23. George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Mateo, 1995. Morgan Kaufmann.
24. Ali Khalili and Sören Auer. Rdface: The rdfa content editor. ISWC 2011 demo track, 2011.
25. J. Kittler, M. Hatef, R. P W Duin, and J. Matas. On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3):226–239, Mar 1998.
26. Ron Kohavi. The power of decision tables. In *8th European Conference on Machine Learning*, pages 174–189. Springer, 1995.
27. Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. *Machine Learning*, 95(1-2):161–205, 2005.
28. S. le Cessie and J.C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.
29. B. W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, 405:442–451, 1975.

30. David Nadeau. Balie—baseline information extraction: Multilingual information extraction from text with machine learning and natural language techniques. Technical report, Technical report, University of Ottawa, 2005.
31. David Nadeau. *Semi-supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision*. PhD thesis, Ottawa, Ont., Canada, Canada, 2007. AAINR49385.
32. David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.
33. David Nadeau, Peter Turney, and Stan Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. pages 266–277, 2006.
34. Axel-Cyrille Ngonga Ngomo, Norman Heino, Klaus Lyko, René Speck, and Martin Kaltenböck. SCMS - Semantifying Content Management Systems. In *Proceedings of the International Semantic Web Conference*, 2011.
35. Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1400–1405. AAAI Press, 2006.
36. J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
37. Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
38. Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and andreas Both. N^3 - A Collection of Datasets for Named Entity Recognition and Disambiguation in the NLP Interchange Format. In *Proceedings of LREC'14*, 2014.
39. G. Sampson. How fully does a machine-usable dictionary cover english text. *Literary and Linguistic Computing*, 4(1), 1989.
40. Robert E. Schapire. The strength of weak learnability. *Mach. Learn.*, 5:197–227, July 1990.
41. Marc Sumner, Eibe Frank, and Mark Hall. Speeding up logistic model tree induction. In *9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 675–683. Springer, 2005.
42. Christine Thielen. An approach to proper name tagging for german. In *In Proceedings of the EACL-95 SIGDAT Workshop*, 1995.
43. D. Walker and R. Amsler. The use of machine-readable dictionaries in sublanguage analysis. *Analysing Language in Restricted Domains*, 1986.
44. Dekai Wu, Grace Ngai, and Marine Carpuat. A stacked, voted, stacked model for named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 200–203, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
45. Pengyi Yang, Yee Hwa Yang, Bing B. Zhou, and Albert Y. Zomaya. A review of ensemble methods in bioinformatics. *Current Bioinformatics*, 5(4):296–308, 2010.
46. GuoDong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of ACL*, pages 473–480, 2002.